GAME THEORETIC DISTRIBUTED COORDINATION: DRIFTING ENVIRONMENTS AND CONSTRAINED COMMUNICATIONS

A Dissertation
Presented to
The Academic Faculty

By

Yusun Lim

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2014

GAME THEORETIC DISTRIBUTED COORDINATION: DRIFTING ENVIRONMENTS AND CONSTRAINED COMMUNICATIONS

Approved by:

Dr. Jeff S. Shamma, Advisor
Professor, School of ECE
Georgia Institute of Technology

Dr. Ayanna MacCalla Howard
Professor, School of ECE
Georgia Institute of Technology

Dr. Yorai Wardi
Professor, School of ECE
Georgia Institute of Technology

Dr. Matthieu Ratoslav Bloch
Asst. Professor, School of ECE
Georgia Institute of Technology

Dr. Faramarz Fekri
Professor, School of ECE
Georgia Institute of Technology

Dr. Eric Feron
Professor, School of AE
Georgia Institute of Technology

Date Approved: July 10, 2014

All of this work is dedicated to my parents - who are the heroes and mentors of my life. I couldn't have done it without you.

ACKNOWLEDGEMENTS

First of all, I appreciate my advisor, Dr. Jeff S. Shamma, for guiding me solve challenging and interesting research problems during my Ph.D. program. I also appreciate my dissertation committees, Dr. Yorai Wardi, Dr. Ayanna MacCalla Howard, Dr. Faramarz Fekri, Dr. Matthieu Ratoslav Bloch, and Dr. Eric Feron.

Also, I would like to thank the lab members for their friendship and meaningful conversations: Dr. Ibrahim al-Shyoukh, Farhan Aziz, Dr. George Chasparis, Dr. Nicolas Dudebout, Dr. Michael J. Fox, Dr. Malachi Jones, Dr. Kwang-Ki Kim, Dr. Georgios Kotsalis, Dr. Lichun Li, Daniel Pickem, Dr. Georgios Piliouras, Sebastian Ruf, Dr. Behrouz Touri, and Ahmet Yazicioglu (sorted by last name).

The work in this thesis was supported by the AFOSR/MURI project (#FA9550-09-1-0538), the ONR project (#N00014-09-1-0751).

Finally my deepest thanks go to my parents and my younger brother, who gave me endless love and support. Their endeavor and support enabled the accomplishment of this dissertation.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

LIST OF TABLES vii

LIST OF FIGURES viii

SUMMARY x

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 BACKGROUND 6
2.1 Game theory 7
2.1.1 Equilibria of a game 7
2.1.2 Game structures 8
2.2 Stochastic stability 11
2.3 Game theoretic learning 13
2.3.1 Payoff-based dynamics 14
2.3.2 Log-linear learning 14
2.3.3 Fictitious play 15
2.3.4 Regret testing 16

CHAPTER 3 STOCHASTIC STABILITY IN DRIFTING ENVIRONMENTS 19
3.1 Introduction 19
3.2 Robustness of stochastic stability 20
3.3 Specialization: Game Theoretic Learning 24
3.4 Example : Sensor coverage problem 27
3.4.1 Simulations 28
3.4.2 Laboratory experiments 29
3.5 Summary 35

CHAPTER 4 COORDINATING WITHOUT COMMUNICATIONS 36
4.1 Introduction 36
4.2 Preliminaries 38
4.2.1 Correlated equilibrium 38
4.2.2 De Bruijn sequences 39
4.2.3 Information theory tools 40
4.3 Team-based zero-sum game 41
4.4 Opponents with limited rationality 43
4.5 Micro-players within a team player 46
4.6 Micro-player matching 49
4.6.1 Algorithm description 51
4.6.2 Simulations 56
4.7 Micro-player matching with a de Bruijn ID sequence 61

4.7.1 Phase difference between action sequences 64
4.7.2 Algorithm description 67
4.7.3 Simulations 72
4.8 Summary and discussion 75
4.8.1 Summary 75
4.8.2 Discussion 76

CHAPTER 5 PAYOFF-BASED COORDINATION IN DRIFTING ENVIRONMENTS 78
5.1 Team-based zero-sum games with adaptive opponents 78
5.1.1 Simulations 79
5.2 Micro-player matching against humans 81
5.2.1 Human testbed program 82
5.2.2 Test results 85
5.3 Summary 87

CHAPTER 6 CONCLUSION AND FUTURE DIRECTIONS 88
6.1 Robustness of stochastic stability 88
6.2 Correlated behavior in payoff-based learning 88
6.3 Future directions 89

REFERENCES 91

LIST OF TABLES

Table 1 Action selection rule for the opponent ($\mathcal{P}_Y$) 57

Table 2 An example of the team's optimal sequences. 57

Table 3 Payoff function $\phi(x^1, x^2, y)$ for simulations 73

LIST OF FIGURES

Figure 1 Relations between game structures 11

Figure 2 State transition diagrams of P^0 and P^ϵ. For P^0, all states are stable. But for P^ϵ, S_1 is the only stochastically stable state. 13

Figure 3 Example of sensor allocation and event density over the mission space . . 28

Figure 4 Event coverage simulation results (a) static events (b) dynamically drifting events. A size of randomly generated noise δ_3 was bounded by 0.0,0.2, or 0.4 for both cases. 29

Figure 5 Snapshots of a dynamic event coverage simulation at every 100 time steps from $t = 3500$ to $t = 4000$. 30

Figure 6 The Khepera III robot. The three small reflective balls are attached to the robot for 3-dimensional position tracking by the OptiTrack system. . 31

Figure 7 The OptiTrack motion capture camera. 32

Figure 8 The structure of the experiment system 32

Figure 9 Event coverage experiment results (a) static events (b) dynamically drifting events 33

Figure 10 Snapshots of a dynamic event coverage experiment at every 10 seconds from 130 seconds to 180 seconds 34

Figure 11 A 3-dimentional De Bruijn graph over a symbol set $\{0, 1\}$. Every Eulerian cycle of the graph is an element of the de Bruijn sequence set $\mathcal{B}(2, 3)$. 40

Figure 12 Simulation result of sequence coordination example 1 57

Figure 13 Simulation results for learning speed comparison. The micro-player matching could coordinate about five times faster than the case in which the team players played payoff-based log-linear learning without introducing a notion of micro-players. 58

Figure 14 Simulation result of sequence coordination example 2 60

Figure 15 Simulation result of micro-player matching with a de Bruijn sequence. The upper graph shows the average reward and the lower one shows the phase difference α. The positive value of α means that $\mathcal{P}_X^1$ is lagging, and a negative value means that $\mathcal{P}_X^2$ is lagging. And α is zero when there is no phase difference between $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$. 74

Figure 16 Team players' average rewards against the adapting opponent. The dotted lines are drawn at $\frac{1}{3}$ which is a maxmin reward of the correlated equilibrium of the game. With the micro-player matching method, the team players could adapt their action sequences against the opponent who adapted its strategy dynamically. 81

Figure 17 Human testbed program. Both *Player 1* and *Player 2* are at a neutral position which is the middle of the *left* and *right* boxes while waiting for the user's input. Once the user selects his action, then each *Player 1* and *Player 2* moves to one of the *left* and *right* boxes according to their actions. 83

Figure 18 Screenshots of the program when a user won and lost the stage. A picture with facial expression helps the user to intuitively recognize a result of each stage. 84

Figure 19 The human testbed program results with three experiment participants. In each graph, the blue line and red line are the average winning rates of the participant against a static action sequence and a team of players with micro-player matching method respectively. 86

SUMMARY

The major objective of this dissertation is extending the capabilities of game theoretic distributed control to more general settings. In particular, we are interested in drifting environments and/or constrained communications.

The first part of the dissertation concerns slowly varying dynamics, i.e., drifting environments. A standard assumption in game theoretic learning is a stationary environment, e.g., the game is fixed. We investigate the case of slow variations and show that for sufficiently slow time variations, the limiting behavior "tracks" the stochastically stable states. Since the analysis is regarding Markov processes, the results could be applied to various game theoretic learning rules. In this research, the results were applied to log-linear learning. A mobile sensor coverage example was tested in both simulation and laboratory experiments.

The second part considers a problem of coordinating team players' actions without any communications in team-based zero-sum games. Generally, some global signalling devices are required for common randomness between players, but communications are very limited or impossible in many practical applications. Instead of learning a one-shot strategy, we let players coordinate a periodic sequence of deterministic actions and put an assumption on opponent's rationality. Since team players' action sequences are periodic and deterministic, common randomness is no longer required to coordinate players. It is proved that if a length of a periodic action sequence is long enough, then opponents with limited rationality cannot recognize its pattern. Because the opponents cannot recognize that the players are playing deterministic actions, the players' behavior looks like a correlated and randomized joint strategy with empirical distribution of their action sequences. Consequently players can coordinate their action sequences without any communications or global signals, and the resulting action sequences have correlated behavior.

Moreover, the notion of micro-players are introduced for efficient learning of long action sequences. Micro-player matching approach provides a new framework that converts the original team-based zero-sum game to a game between micro-players. By introducing a de Bruijn sequence to micro-player matching, we successfully separate the level of opponent's rationality and the size of the game of micro-players. The simulation results are shown to demonstrate the performance of micro-player matching methods.

Lastly, the results of the previous two topics are combined by considering a problem of coordinating actions without communications in drifting environments. More specifically, it is assumed that the opponent player in the team-based zero-sum games tries to adjust its strategy in the set of bounded recall strategies. Then the time-varying opponent's strategy can be considered as a dynamic environment parameter in a coordination game between the team players. Additionally, we develop a human testbed program for further study regarding a human as an adaptive opponent in the team-based zero-sum games. The developed human testbed program can be a starting point for studying game theoretic correlated behavior learning against a human.

CHAPTER 1

INTRODUCTION

Multi-agent systems have been received tremendous attention for their capability to solve problems that are impossible or challenging for a single agent. Compared with single-agent systems and centralized approaches, multi-agent systems provide a more natural framework for many problems such as resource allocation and team planning. Moreover multi-agent systems are robust in the sense that if an agent in a system has broken or failed then other agents can still maintain the system. A precise definition of 'agents' in multi-agent systems can be a little bit different for different researchers. But generally, entities that autonomously make local decisions and interact with other agents and/or environments are considered as agents. Robots, humans or even software without a physical body can be agents for a multi-agent system.

One of the most important characteristics of multi-agent systems is the absence of a centralized structure. Even though a system may have multiple entities, if there exists a designated one that can observe the system's global state and control the global behavior, then the system can be considered a single-agent system. Therefore control algorithms for multi-agent systems are inherently distributed control.

Coordinating a team of agents to achieve collective goals is usually a major objective of control design for multi-agent systems. However designing such control without centralized structures is generally more difficult than centralized system design. An agent's decision is based on local information and it influences directly and indirectly other agents. It may result in undesired global behaviors and increase the complexity of the system.

Game theory has received considerable attention as a new paradigm for multi-agent systems and distributed control because of its many advantages, such as minimal communication requirements, robustness to failures and environmental disturbances, and scalability [1]. Of specific interest in the game theory literature is game theoretic learning

[2, 3, 4]. Game theoretic approaches model a system as a non-cooperative game between self-interested agents, and establish simple algorithms that converge to the emergent equilibrium of the game with desired global behavior. The relevance to engineered systems is that distributed components can be incentivized through the appropriate design of a game whose equilibria represent a desirable operating condition. A representative reference is [5] which presents connections between game theoretic learning and various cooperative control problems. There exists a rich set of applications based on game theoretic designs, including mobile sensor coverage problems [6], wind energy harvesting [7], vehicle-target assignment [8], power management in sensor networks [9] and ad hoc networks [10].

The major objective of this research is extending the capabilities of game theoretic distributed control to more general settings. In particular, we are interested in drifting environments and/or constrained communications.

First, we consider game theoretic learning in drifting environments. The standard game theoretic learning assumes that environments in which agents are interacting each others are not time-varying. Therefore, in such static environments, the same joint action of agents always gives the same results regardless of the time when the actions are played. However there are many engineering systems with dynamic and time dependent environments, and analyzing game theoretic learnings in such environments is essential for these settings.

We define 'drifting environments' as a type of dynamic environment whose changing speed is limited and slow. It is shown that a stochastically stable state, which is a notion of stability in game theory literature, is also drifting with limited speed in such drifting environments. We proved that game theoretic learnings can track the drifting stochastically stable states if their drifting speed is slow enough. The number of existing learning algorithms in the literature is large, so analyzing each one of those algorithms one-by-one is unrealistic. Instead, we analyze robustness of stochastic stability of Markov process within a drifting environment. Because many adaptive learning algorithms can be modeled as

Markov processes, Markov process based analysis validates performance of various learning algorithms. In this thesis, log-linear learning [11, 5, 12] was analyzed under the drifting environment setting as an illustrated example. Then the results were tested with a sensor coverage problem in both simulations and laboratory experiments with mobile robots.

The second part of this thesis considers the problem of coordinating a team of agents without communications. In general cooperative multi-agent control settings, agents communicate with each other to exchange their local information. Many difficulties in designing multi-agent systems are inherited from asymmetry of information among agents. Hence how and what information is shared among agents is one of the most important concerns in multi-agent systems. If agents could communicate information without any cost and delay then all information in the system would be shared globally. In this case, one can apply a centralized control to coordinate agents in the system. Unfortunately, communications between agents are not free in many multi-agent systems. It often requires time and resources, and sometimes it is even impossible because of various practical reasons.

Another problem concerning communication is a security issue under the existence of hostile opponents or environments. There are many strategic situations in which a group of players has to coordinate their plans while these are hidden from outside of the group. In baseball games, a pitcher and a catcher communicate through a series of hand signals and it is very important to make sure that the hitter is not stealing their signs. Allied armies in a war utilize cryptography to keep their coordinated plans secret from enemies. Even though messages are encrypted in such situations, communication always has a security risk as long as there are hostile opponents that can take advantage of the communicated information.

The main idea of coordinating players without communications is adapting a player's periodic action sequence instead of a randomized one-shot action strategy. Globally observable signals or communication between players are required because common randomness is needed for coordination. If players' actions are deterministic periodic sequences so that

common randomness is not needed, then players can coordinate their joint actions without communications. We put an assumption on opponents' rationality and it is proved that a pattern of players' actions is not recognized by such opponents. However learning a series of actions instead of one mixed action is generally very slow because of the increased size of the space in which the learning process is searching the optimal point.

To adapt a long action sequence efficiently, we introduce a notion of micro-players and micro-player matching methods. Micro-players are virtual entities within an actual player who wants to adapt a periodic action sequence. Like general players, each micro-player has its own internal states and adapts its one-shot action whenever it is selected by the actual player who contains it. Then the actual player can play a periodic action sequence by selecting a micro-player with a periodic order. In micro-player matching, all players have the same periodic order of micro-players. Then the original game between players is converted to a corresponding game among micro-players.

Lastly, this thesis joins the previous two topics together by considering game theoretic learnings in drifting environments without communications. A team of players coordinates their actions without communications and their opponent also adapts their own strategies. The opponent who dynamically adapts its strategy plays a role as dynamic environments from the team players' point of view. If the opponent's learning speed is slow enough relative to the team players learning, then the robustness result of stochastic stability can be applied.

Additionally, we suggest an interesting research direction that considers a human as the limited-rationality opponent. The exact learning mechanism of humans is not understood yet, and understanding it is beyond the scope of this research. But apparently, a human is also an agent who learns and adapts his/her actions with limited rationality. Therefore we believe that testing our algorithms against human participants is worthwhile. A graphical user interface is developed as a human testbed program for experiments with human participants. With the developed testbed program, experiment participants play a game against a

team of virtual players coordinating their actions using micro-player matching.

This thesis is organized as follows. Chapter 2 presents some important background for game theoretic learning. Mathematical notations are also defined in this chapter. Then game theoretic learnings in drifting environments and limited communication are separately studied in Chapter 3 and Chapter 4. Chapter 5 combines those results. Finally, Chapter 6 presents concluding remarks.

CHAPTER 2

BACKGROUND

A major goal of distributed control is developing underlying theories for design and control of multi-agent systems in which interacting subsystems make local decisions based on local information. Game theory from economics concerns mathematical models of interacting decision makers, hence one can easily find the relevance between distributed control and game theory. However many game theoretic results from the economic literature cannot be directly applied to distributed control in engineering. A major reason is that game theory analyzes existing social systems from a descriptive point of view but engineering system design requires a prescriptive framework [1].

Some recent research such as [1] and [13] provide an architectural view that adopts game theory as a prescriptive tool for distributed control system design. The proposed architecture in [1, 13] modularizes game theoretic control into two components: (i) utility design which defines individual agents' local interest, and (ii) learning design which defines local decision making rules. This decoupling can be obtained by enforcing a certain structure on the game [1]. The enforced structure becomes an interface between utility design and learning design, and simplifies design processes. The interface requires utility design to ensure that the resulting game has a specific structure and requires learning design to provide desired behavior when it is applied to the game structure.

This modularization allows a general design framework rather than application-specific designs. Distributed control problems involve various constraint settings, and developing designs subject to such application-specific constraints is generally a difficult task. By this modularized architectural view, application system designers would choose a specific utility design and learning design from a rich set of developed general designs according to constraints of the application.

The rest of this chapter will present game theoretic background related to this research,

as well as stochastic stability and some game theoretic learning rules.

2.1 Game theory

2.1.1 Equilibria of a game

Let $\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2, ..., \mathcal{P}^n\}$ be a set of n-players, and $\mathcal{A}_i$ be a set of available actions for each player i. This research will assume that $\mathcal{P}$ and $\mathcal{A}_i$ are always finite. Let $\mathcal{A}$ be the set of joint actions, i.e., $\mathcal{A} = \prod_{i \in N} \mathcal{A}_i$. The payoff function (or utility function) for each player $\mathcal{P}^i$ is $U_i : \mathcal{A} \to \mathcal{R}$. Then one can define a game $\mathbf{G}$ as a tuple $\mathbf{G} = \langle \mathcal{P}, \{\mathcal{A}_i\}_{i \in \mathcal{P}}, \{U_i\}_{i \in \mathcal{P}} \rangle$. For convenience, let a_{-i} be the collection of actions of players other than $\mathcal{P}^i$, i.e.,

$$a_{-i} = (a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_n).$$

A Nash equilibrium is the most central solution concepts in non-cooperative games.

Definition 2.1.1. *An action profile $a^* \in \mathcal{A}$ is a pure strategy Nash equilibrium if for all player $\mathcal{P}^i$,*

$$U_i(a_i^*, a_{-i}^*) = \max_{a_i \in \mathcal{A}_i} U_i(a_i, a_{-i}^*).$$

A Nash equilibrium is a stable strategy in the sense that no player has an incentive to change his action unilaterally from a current action profile. Game can have zero, one or multiple pure Nash equilibria. Let $\Delta(\mathcal{A}_i)$ be the set of all probability vectors over $\mathcal{A}_i$. If players are allowed to randomize or 'mix' his actions then every game $\mathbf{G}$ has at least one Nash equilibrium. A mixed strategy $p_i \in \Delta(\mathcal{A}_i)$ stochastically selects a pure action with a fixed probability p_i.

Even though a Nash equilibrium is a result based on the logic that each player tries to maximizing his own welfare, it may not be the 'optimal' action profile for the game from the point of an outside observer's view. A prisoner's dilemma is a classic example of a non-optimal Nash equilibrium. Consider a set of player $\mathcal{P} = \{\mathcal{P}^1, \mathcal{P}^2\}$ with an action set $\mathcal{A} = \{\{C, D\}, \{C, D\}\}$. In other words, each player can choose either *Cooperate* or *Defect*. The following matrix defines the utility of each players.

	C	D
C	3,3	0,4
D	4,0	1,1

The first player $\mathcal{P}^1$ selects row and the second player $\mathcal{P}^2$ selects column. Then the first elements of the selected pair is the utility for $\mathcal{P}^1$ and the second elements is the utility for $\mathcal{P}^2$. Considering the average utility, the joint action (C, C) yields the best result for the players. However if one player decides to cooperate, then another player can increase his utility by choosing to defect. Hence (C, C) is not a Nash equilibrium, and the only Nash equilibrium of the game is (D, D) which receives the lowest average utility.

Efficiency of equilibrium can be measured by price of anarchy (PoA) and price of stability (PoS) [14]. Define global welfare as $\phi : \mathcal{A} \to \mathcal{R}$. Let $\mathcal{E}(\mathbf{G})$ denote the set of equilibrium of a game $\mathbf{G}$. Then price of anarchy (PoA) and price of stability (PoS) are defined as follows.

$$\text{PoA}(\mathbf{G}) = \min_{a^{\text{ne}} \in \mathcal{E}(\mathbf{G})} \frac{\phi(a^{\text{ne}})}{\phi(a^{\text{opt}})}$$

$$\text{PoS}(\mathbf{G}) = \max_{a^{\text{ne}} \in \mathcal{E}(\mathbf{G})} \frac{\phi(a^{\text{ne}})}{\phi(a^{\text{opt}})}$$

where $a^{\text{NE}} \in \mathcal{A}$ is a Nash equilibrium and $a^{\text{opt}} \in \arg\max_{a^* \in \mathcal{A}} \phi(a^*)$ is an optimal joint action profile. Hence PoA and PoS give a lower bound and a upper bound on the global welfare achieved by any Nash equilibrium respectively. In general, the price of anarchy can be arbitrarily close to 0. If there exists more than one equilibrium, selecting a equilibrium with larger global welfare is an important consideration when designing learning rules.

2.1.2 Game structures

We will introduce the following three classes of games structures; identical interest games, potential games, and weakly acyclic games. Generally in engineered multi-agent systems, there exists a global objective function $\phi : \mathcal{A} \to \mathcal{R}$ that system designers want to maximize. In such systems, it is important to align individual players' utility functions with a global

objective function. Note that the key difference between these classes is the degree of the alignment between players' utility functions and the global objective function.

In identical interest games, each player's local utility function is equal to a global objective function. Identical interest games can be considered as one of the most restrictive form of games for cooperative control. Local agents in many applications are not able to measure global welfares, and such application systems cannot be modeled as identical interest games.

Definition 2.1.2. *A game,* $\mathbf{G}$*, is an identical interest game if it satisfies the following property for some function* $\phi : \mathcal{A} \to \mathbb{R}$

$$U_i(a) = \phi(a)$$

for every player $\mathcal{P}^i$ *and for every joint action* $a \in \mathcal{A}$*.*

It is trivial to verify that every identical interest game has at least one pure Nash equilibrium and any action profile a maximizing ϕ is a Nash equilibrium.

The class of potential games [15], which is a generalization of identical interest games, has been considered as one of the most powerful interfaces for modularization. In potential games, each agent's local interest is not exactly same as a global objective, but aligned with it.

Definition 2.1.3. *A game* G*, is a potential game [15] if it satisfies the following property. There exists a potential function* $\phi : \mathcal{A} \to \mathbb{R}$*, such that for all* $a = (a_i, a_{-i}) \in \mathcal{A}$ *and* $a' = (a'_i, a_{-i}) \in \mathcal{A}$*,*

$$\phi(a_i, a_{-i}) - \phi(a'_i, a_{-i}) = U_i(a_i, a_{-i}) - U_i(a'_i, a_{-i}).$$

Every potential game has at least one pure Nash equilibria. Connection between cooperative control and potential games is well studied by [5, 16]. There exist lots of learning algorithms for potential games such as Joint Strategy Fictitious Play (JSFP) [17], log-linear learning[11, 5, 12], dynamic fictitious play [18], and others [19, 20, 21].

Marginal contribution utility, which is sometimes referred to as Wonderful Life Utility (WLU) [22], is one of promising utility designs that lead to a potential game. WLU is a player's marginal contribution to a global objective, i.e.,

$$U_i(a_i, a_{-i}) = \phi(a_i, a_{-i}) - \phi(a_i^{\emptyset}, a_{-i})$$

where $a_i^{\emptyset}$ indicates a null action.

Weakly acyclic games [23, 4] are a more general class of potential games in the sense that it requires less structure. To explain a definition of weakly acyclic games, consider a game $\mathbf{G}$ with better reply dynamics as follows. Suppose that only one player is allowed to move at a time, and the player moves in a way to increase his own utility. Let $\mathcal{P}^{i(t)}$ be a player who is selected to move at time t. Then a better reply path is a sequence of action profile $a(1), a(2), a(3), ...a(T)$ that, for every index $t < T$, $a_{i(t)}(t) \neq a_{i(t)}(t+1)$, $a_{-i(t)}(t) = a_{-i(t)}(t+1)$, and $U_{i(t)}(a(t)) < U_{i(t)}(a(t+1))$. The definition of a weakly acyclic game is as follows.

Definition 2.1.4. *A game* $\mathbf{G}$ *is a weakly acyclic game if for every* $a \in \mathcal{A}$*, there exists a better reply path starting at* a *and ending at some pure Nash equilibrium of* $\mathbf{G}$.

The following proposition identifies the similarities between potential games and weakly acyclic games.

Proposition 2.1.1. *A game* $\mathbf{G}$ *is weakly acyclic if and only if there exists a potential function* $\phi : \mathcal{A} \to \mathbb{R}$ *such that for any action* $a \in \mathcal{A}$ *that is not a Nash equilibrium, there exists a player* $\mathcal{P}^i$ *with an action* $a_i' \in \mathcal{A}_i$ *such that*

$$U_i(a_i, a_{-i}) - U_i(a_i', a_{-i}) \to \phi(a_i, a_{-i}) - \phi(a_i', a_{-i}).$$

This class of games captures various problems in cooperative games, such as congestion games, distributed routing in networks, and area coverage [19]. Many learning algorithms such as [19, 21, 4] are developed for weakly acyclic games. Note that identical interest games and potential games are special cases of weakly acyclic games. Figure 1 illustrates relations between game structures.

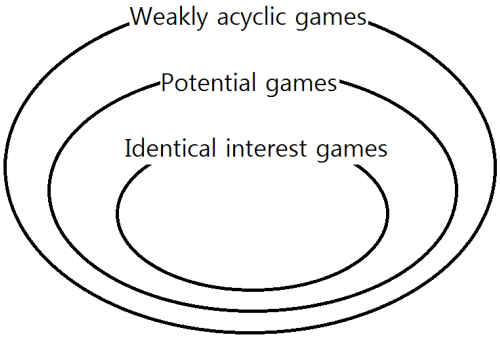


Figure 1: Relations between game structures

2.2 Stochastic stability

Game theoretic learning concerns adaptive learning rules and their convergence to equilibrium concepts such as Nash equilibrium. One of the important convergence concepts in game theoretic learning is stochastic stability which was originally introduced in [20].

For $x \in \mathcal{R}^n$, $|x|$ denotes the usual Euclidean norm, and $\|M\|$ denotes the associated induced matrix norm. The vector $\mathbf{1}$ denotes a column vector $\begin{pmatrix} 1 & 1 & ... & 1 \end{pmatrix}^{\mathrm{T}}$ of appropriate context dependent dimension. $|Z|$ denotes the number of elements of the finite set, $Z = \{1, 2, ..., n\}$. $\Delta(n)$ denotes the n-dimensional probability simplex, i.e.,

$$\Delta(n) = \left\{ x \in \mathbb{R}^n \;\middle|\; \mathbf{1}^{\mathrm{T}} x = 1, x_i \geq 0 \right\}.$$

Let $\mathbb{S}_n$ denote the set of all $n \times n$ stochastic matrices, i.e., matrices with non-negative entries whose rows sum to one. Any $P \in \mathbb{S}_n$ defines a Markov chain over some state space Z with $|Z| = n$, where the ij^{th} element of P, P_{ij}, denotes the transition probability from state $i \in Z$ to state $j \in Z$. In case P has a single aperiodic recurrent class [24], there exists a unique stationary distribution $\mu \in \Delta(n)$ such that

$$\mu^{\mathrm{T}} P = \mu$$

and for all $x \in \Delta(n)$,

$$\lim_{t \to \infty} x^{\mathrm{T}} P^t = \mu^{\mathrm{T}}.$$

The following definitions are adopted from the presentations in [20] and [23].

Definition 2.2.1. *Let $P^0 \in \mathbb{S}_n$ define a Markov chain on $Z = \{1, 2, ..., n\}$. Let $\left\{ P^\epsilon \;\middle|\; \epsilon \in (0, \bar{\epsilon}] \right\} \subset \mathbb{S}_n$ define a family Markov chains on Z. The family $\{P^\epsilon\}$ is a **regular perturbation** of P^0 if i) each P^ϵ has a single aperiodic recurrent class; ii) for each $i, j \in Z$,*

$$\lim_{\epsilon \to 0} P^\epsilon_{ij} = P^0_{ij};$$

and iii) there exists a $\mu^* \in \Delta(n)$ such that

$$\lim_{\epsilon \to 0} \mu^\epsilon = \mu^*,$$

where μ^ϵ is the unique stationary distribution of P^ϵ.

Definition 2.2.2. *Let $\{P^\epsilon\}$ be a regular perturbation of P^0. A state $i \in Z$ is **stochastically stable** if*

$$\lim_{\epsilon \to 0} \mu^\epsilon_i = \mu^*_i > 0.$$

The interpretation is that a stochastically stable state has non-vanishing occupancy frequency for small ϵ as $t \to \infty$ and other states will be almost vanished. As an intuitive example, consider a three state Markov process P^0. Let $\{S_1, S_2, S_3\}$ be a set of states and the state transition matrix be

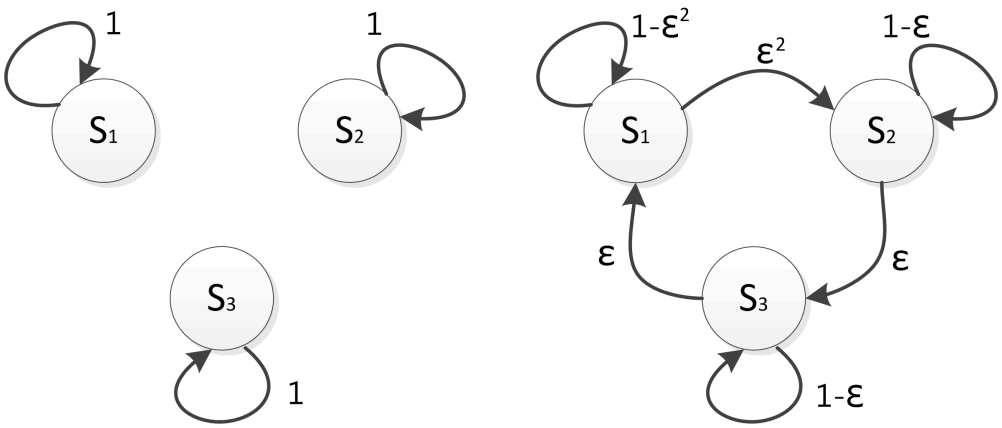
$$P^0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

It is obvious that every state is stationary in this system.

Now consider a regular perturbed Markov process P^ϵ.

$$P^\epsilon = \begin{pmatrix} 1-\epsilon^2 & \epsilon^2 & 0 \\ 0 & 1-\epsilon & \epsilon \\ \epsilon & 0 & 1-\epsilon \end{pmatrix}.$$

Figure 2 illustrates state transition diagrams of P^0 and P^ϵ. Assume the value ϵ is very small. Then a probability that a state deviates from S_2 or S_3 is ϵ while a probability deviating from



(a) Markov chain P^0 (b) P^ϵ which is a regular perturbation of P^0

Figure 2: State transition diagrams of P^0 and P^ϵ. For P^0, all states are stable. But for P^ϵ, S_1 is the only stochastically stable state.

S_1 is ϵ^2 which is much smaller than ϵ. One can easily expect that a state of the system will spend most of times at S_1 after a time long enough, and occupancy frequency of S_2 and S_3 will goes to zero as ϵ converges to zero. Therefore S_1 is the only stochastically stable state of the system. The robustness of stochastic stability will be analyzed in Chapter 3.

2.3 Game theoretic learning

This section introduces some game theoretic learning algorithms related to this research. Game theoretic learning is about how players can learn to play desired joint action profiles such as a Nash equilibrium through repeated interactions [2, 25, 26, 27]. For repeated game settings, let $a_i(t)$ and $p_i(t)$ denote the realized action and mixed strategy of $\mathcal{P}^i$ at time t. All learning algorithms discussed in this thesis are based on the following process. At each time $t = 1, 2, 3, ...$, each player $\mathcal{P}^1$ simultaneously selects its action $a_i(t) \in \mathcal{A}_i$ with a probability $p_i(t) \in \Delta(\mathcal{A}_i)$. Then it receives its utility $U_i(a_i(t), a_{-i}(t))$ and adjust its strategy p_i using local information. A range of local information which is accessible by a player may vary based on the system's attributes. Game theoretic learning rules define the strategy adjustment mechanism of players.

Because of the large number of existing learning algorithms in this literature, learning

algorithms introduced here are only a small part of the entire game theoretic learnings. One can refer to other references like [2, 4, 28] for more kinds of learning algorithms.

2.3.1 Payoff-based dynamics

One of the criteria to categorize general game theoretic learning rules is the information dependency of individual player's learning rules. A learning rule is called 'uncoupled' if a player's strategy does not depend on other players' payoffs. It is called 'radically uncoupled' or 'payoff-based' if a player's strategy does not depend on other players' payoffs and actions.

The payoff based setting is considered as the most restrictive observational condition for agents in multi-agent systems [19]. In a payoff based scenario, no form of communication is allowed, and a player only has information about the action it played and (possibly noisy) the reward it received. A player with a payoff-based learning is not aware of the presence of other players so that he has to adapt its strategy only based on history of its own realized payoffs and actions. In many practical systems, the precise payoff structure is unknown to players and a player cannot observe actions of other players. Therefore this 'payoff-based' condition is often a reasonable requirement when designing learning rules. There are payoff based learning rules like [19, 12, 29, 30] in game theoretic control literature.

Developing payoff based learning rules that converge to correlated strategies is not trivial because of the lack of correlating devices such as communications or global signal devices. Chapter 4 demonstrates a payoff-based method to achieve performance close to correlated strategies without communications between players.

2.3.2 Log-linear learning

Log-linear learning, introduced in [11] and further developed in [5, 12], is a learning algorithm for potential games. Some variants of standard log-linear learning such as binary log-linear learning and payoff-based log-linear learning are employed in this proposal. The algorithm of log-linear learning proceeds as follows. At each stage $t > 0$, a single agent

$\mathcal{P}^i \in \mathcal{P}$ is randomly chosen and allowed to alter his current action. All other players must repeat their actions from the previous stage, i.e. $a_{-i}(t) = a_{-i}(t-1)$. At stage t, the selected player i employs the Boltzmann distribution strategy $p_i(t) \in \Delta(\mathcal{A}_i)$, given by

$$p_i^{a_i}(t) = \frac{e^{\frac{1}{\tau}U_i(a_i, a_{-i}(t-1))}}{\sum\limits_{\bar{a}_i \in \mathcal{A}_i} e^{\frac{1}{\tau}U_i(\bar{a}_i, a_{-i}(t-1))}}, \tag{1}$$

for a fixed "temperature" parameter, $\tau > 0$. As is well known for the Boltzmann distribution, for large τ, player i will select any action $a_i \in \mathcal{A}_i$ with approximately equal probability, whereas for diminishing τ, player i will select a best response to the action profile $a_{-i}(t-1)$, i.e.,

$$a_i(t) \in \arg\max_{a_i \in \mathcal{A}_i} U_i(a_i, a_{-i}(t-1))$$

with increasingly high probability.

The following theorem characterizes the limiting behavior associated with log-linear learning for the class of potential games.

Theorem 2.3.1 (Blume, 1993 [11])**.** *Let G be a finite n-player potential game. Log-linear learning induces an aperiodic and irreducible process of the joint action set $\mathcal{A}$. Furthermore, the unique stationary distribution $\mu(\tau) = \{\mu^a(\tau)\}_{a \in \mathcal{A}} \in \Delta(\mathcal{A})$ is given by*

$$\mu_a^\tau = \frac{e^{\frac{1}{\tau}\phi(a)}}{\sum\limits_{\bar{a} \in \mathcal{A}} e^{\frac{1}{\tau}\phi(\bar{a})}}. \tag{2}$$

The explicit characterization of the stationary distribution in (2) reveals that as $\tau \to 0$, all of the weight of the stationary distribution is on the set $\left\{a^* \in \mathcal{A} \;\middle|\; \max_{a \in \mathcal{A}} \phi(a) = \phi(a^*)\right\}$, i.e., the set of potential function maximizers.

2.3.3 Fictitious play

Fictitious play is one of well-known decision making rules in game theory. Roughly speaking, a player with fictitious play assumes that other players are playing stationary strategies

and estimates other players' strategies using empirical frequency of other players actions during the past time steps. Let a empirical frequency, $q_i(t) \in \Delta(\mathcal{A}_i)$, of $\mathcal{P}^i$ be a running average of the actions of $\mathcal{P}^i$. In fictitious play, the strategy of $\mathcal{P}^i$ at time t is the best response to the other players empirical frequency, i.e.,

$$p_i(t) = \arg\max_{p_i \in \Delta(\mathcal{A}_i)} U_i(p_i, q_{-i}(t)).$$

Joint Strategy Fictitious Play (JSFP) [17], which is a variant of fictitious play, adds notion of 'fading memory' and 'inertia' to fictitious play. Fading memory exponentially discounts influence of past results, and inertia balances a probability to optimize its action and willingness to maintain previous strategy. The reference [17] proved that for potential games, if all players adhere to joint strategy fictitious play, then their joint action profile converges almost surely to a pure Nash equilibrium of the game.

2.3.4 Regret testing

Regret testing is one of payoff-based learning rules introduced by [29]. Because it is a payoff-based learning rule, an individual learning rule of regret testing depends only on summary statistics of a player's own realized payoffs. Suppose that each player $\mathcal{P}^i$ has an finite action set $\mathcal{A}_i$. Assume that there are 'hats' that contain $h_i > |\mathcal{A}_i|$ 'tickets' on which the actions are written. A hat is a device for generating probability distributions over $\mathcal{A}_i$, and the player's strategy can be described as selecting 'a ticket' within 'a hat.'

Let $\mathbf{G}$ be a finite two-person game then a regret testing rule repeats the following process. A period consists of s steps, where s is a large number. For each period, one hat is selected as a player $\mathcal{P}^i$'s base strategy. Every time step during the period, he selects a ticket from its current hat and takes the action written on the selected ticket. After playing the action, the ticket is returned to the hat. At random time steps with probability ω_i, a player experiments a uniformly randomly chosen action instead of drawing a ticket from the hat. Whenever a player plays an action, he receives a payoff. At the end of the period, he calculates the average payoff $\bar{u}$ for playing actions from the hat. And he also tallies the

average payoff $\bar{u}_j$ for playing an experimented action $a_j \in A$. For each action j, let the estimated regret $\hat{r}_j$ be a difference between $\bar{u}_j$ and $\bar{u}$, i.e.,

$$\hat{r}_j = \bar{u}_j - \bar{u}.$$

In other words, $\hat{r}_j$ is a statistical estimate of payoff loss for playing a strategy from the hat instead of playing an action a_j. If $\hat{r}_j$ is greater than his tolerance level $\tau_i > 0$ for at least one action a_j, then he chooses a new hat for the next period. Each hat has a positive probability to be chosen when he choosing a new hat. Otherwise, he repeats the process with his current hat during the next period.

This procedure is called a regret testing because of the logic behind it. If a player has a significantly large amount of regret for at least one action, the player is not satisfied by his current hat and chooses a new hat, i.e., a new strategy for the next period.

Let a joint strategy be an ϵ-Nash equilibrium of $\mathbf{G}$ if no player can increase his payoff by more than ϵ through a unilateral change of strategy. Then the following theorem is proven by [29].

Theorem 2.3.2 (Foster and Young, 2006 [29])**.** *Let* $\mathbf{G}$ *be a finite two-person game played by regret testers and let* $\epsilon > 0$*. There are upper bounds on the tolerance* τ_i *and exploration rate* ω_i*, and lower bounds on* h_i *and frequency of play* s*, such that, at all sufficiently large times* t*, the players' joint behavior at* t *constitutes an* ϵ*-Nash equilibrium of* $\mathbf{G}$ *with probability at least* $1 - \epsilon$*.*

The regret testing rule is extended to the cases with more than two players in [31] . The difference between the dynamics from [31] and the regret testing illustrated above is that there is a small probability λ_i that a player $\mathcal{P}^i$ selects a new action even if no experimented action has a regret more than a tolerance τ_i. This ensures that there always some small probability to experiment a new action for all players throughout the entire learning process.

Theorem 2.3.3 (Germano and Lugosi, 2007 [31])**.** *Let* $\mathbf{G}$ *be a finite* n*-person game played by regret testers and let* $\epsilon > 0$*. There are upper bounds on the tolerance* τ_i*, exploration rate*

ω_i and updating rate λ_i, and lower bounds on h_i and frequency of play s, such that, at all sufficiently large times t, the players' joint behavior at t constitutes an ϵ-Nash equilibrium of $\mathbf{G}$ with probability at least $1 - \epsilon$.

In Chapter 4, this variant of regret testing learning rule from [31] is used to optimize each elements of action sequences.

CHAPTER 3

STOCHASTIC STABILITY IN DRIFTING ENVIRONMENTS

3.1 Introduction

A distributed adaptation rule may have multiple convergent outcomes. In assessing stochastic stability, one analyzes a randomized perturbation of this rule. The effect of the randomization is to induce mixing among all possible steady state outcomes. However, as the perturbation size is diminished, it may be that one outcome is favored among others in that in the long run it occurs with increasing probability whereas other outcomes occur with vanishingly small probability as the size of the randomized perturbation diminishes. In other words, a stochastically stable outcome occupies almost all of the mass of an associated underlying stationary distribution.

Two standard references are [32, 20], which discuss equilibrium selection between risk dominant or payoff dominant equilibrium in symmetric 2×2 games. Their analysis examines a particular stochastic adaptation rule and establishes that risk dominant Nash equlibria are stochastically stable over payoff dominant equilibria. Other utilizations of stochastic stability include a simple model of language evolution [33], and emergent structures in self-assembly [34].

This chapter investigates the robustness of stochastic stability[1], i.e., how stochastically stable states are affected by deviations from a nominal setup. A standard assumption in game theoretic learning is a stationary environment, e.g., the game is fixed throughout the entire process. We investigate the case of slow variations and show that, for sufficiently slow time variations, the limiting behavior "tracks" the stochastically stable states.

Other papers also have examined robustness in game theoretic learning rules. These works include modifications of log-linear learning [12], perturbations of so-called potential games [36, 37], and variations on player selection and tie-breaking rules [38, 39].

[1]The results described in this chapter appear in [35]

This chapter is organized as follows. Section 3.2 presents the main results regarding robustness of stochastic stability. Section 3.3 specializes these results to learning in games, and in particular, to binary log-linear learning. The results are applied to a sensor coverage problem as an illustrative example in Section 3.4. Both simulations and laboratory experiments illustrates the results for coverage games. Finally, the last section presents concluding remarks.

3.2 Robustness of stochastic stability

Consider a perturbed stochastic system in the following form. Let $\mathbb{S}_n$ denote the set of all $n \times n$ stochastic matrices. At each time step $t = 0, 1, 2, ...$, there exists a selection $P(t) \in \mathbb{S}_n$. A state $z(t)$ in a state set $Z = \{1, 2, ..., n\}$ evolves with the Markov property as follows.

$$\begin{aligned} &\mathbf{Pr}\left[z(t+1) \,\middle|\, z(0), z(1), ..., z(t)\right] \\ &= \mathbf{Pr}\left[z(t+1) \,\middle|\, z(t)\right] = P_{z(t)z(t+1)}(t). \end{aligned} \tag{3}$$

In other words, there is a set of possible transition probabilities at each stage, and the transition probabilities at stage t observe the Markov property with transition probabilities determined by the specific selection $P(t)$.

Now consider a stochastic matrix $P^* \in \mathbb{S}_n$, and assume that $P(t)$ is selected from a neighborhood of P^*. Or it can be interpreted as that $P(t)$ is a linear system P^* with some small noise, and $||P(t) - P^*||$ characterizes a size of the noise. Then the following theorem proves a robustness of stochastic stability of the noisy process $P(t)$.

Theorem 3.2.1 (Lim & Shamma, 2013 [35])**.** *Let $P^* \in \mathbb{S}_n$ have a single aperiodic recurrent class, and let $\mu^* \in \Delta(n)$ be the associated stationary distribution over finite state space Z. For any $\delta_1 > 0$, there exists a $\delta_2 > 0$ such that for the dynamic process defined*

by (3),

$$\limsup_{t=0,1,2,...} \|P(t) - P^*\| < \delta_2$$

$$\Rightarrow$$

$$\limsup_{t=0,1,2,...} |\mathbf{Pr}\left[z(t) = i\right] - \mu_i^*| < \delta_1, \textit{ for all } i \in Z.$$

Proof. Let the initial condition $z(0)$ be selected according to the probability distribution $x(0) \in \Delta(n)$, and define the switching linear system

$$x(t+1) = P^{\mathrm{T}}(t)x(t).$$

By construction, for any $i \in Z$ and $t = 0, 1, 2, ...$, $\mathbf{Pr}\left[z(t) = i\right] = x_i(t)$. Let W be an $n \times (n-1)$ matrix with orthonormal columns (i.e., $W^{\mathrm{T}}W = I$) that span the nullspace of $\mathbf{1}^{\mathrm{T}}$. Since $x(t)$ evolves over the simplex, one can uniquely write

$$x(t) = \mu^* + Ww(t)$$

for some $w(t) \in \mathbb{R}^{n-1}$. Define $E(t) = P(t) - P^*$. Then the above linear iterations can be written as

$$\begin{aligned} x(t+1) &= P^{\mathrm{T}}(t)x(t) \\ &= (P^{*\mathrm{T}} + E^{\mathrm{T}}(t))(\mu^* + Ww(t)) \\ &= P^{*\mathrm{T}}\mu^* + E^{\mathrm{T}}(t)\mu^* + P^{*\mathrm{T}}Ww(t) + E^{\mathrm{T}}(t)Ww(t). \end{aligned}$$

Since μ^* is a stationary distribution of P^*, $P^{*\mathrm{T}}\mu^* = \mu^*$. Accordingly,

$$(x(t+1) - \mu^*) = P^{*\mathrm{T}}Ww(t) + E^{\mathrm{T}}(t)\mu^* + E^{\mathrm{T}}(t)Ww(t)$$

$$\Rightarrow$$

$$w(t+1) = \Big(W^{\mathrm{T}}P^{*\mathrm{T}}W + W^{\mathrm{T}}E^{\mathrm{T}}(t)W\Big)w(t) + W^{\mathrm{T}}E^{\mathrm{T}}(t)\mu^*.$$

Let $A = W^{\mathrm{T}}P^{*\mathrm{T}}W$. Since P^* has a single aperiodic recurrent class, A is a stability matrix. In particular, there exists a positive definite $X = X^{\mathrm{T}}$ such that

$$A^{\mathrm{T}}XA - X < 0.$$

Standard Lyapunov arguments establish that $\limsup_{t=0,1,2,...} |w(t)|$ can be made arbitrarily small, since diminishing δ_2 implies diminishing $\|E(t)\|$. □

Note that the above theorem does not assume anything about the structure of the system noise $(P(t)-P^*)$ except its maximum size. Theorem 3.2.1 can be combined with the notion of stochastic stability.

Corollary 3.2.1 (Lim & Shamma, 2013 [35]). *Let $\{P_\epsilon\}$ be a regular perturbation of P^0, and let μ^* be the associated distribution characterizing stochastic stability. Let the dynamic process (3) satisfy*

$$\limsup_{t=0,1,2...} \|P(t) - P^\epsilon\| < \delta_2.$$

For any $\delta_1 > 0$, there exist $\epsilon > 0$ and $\delta_2 > 0$ such that

$$\limsup_{t=0,1,2,...} |\mathbf{Pr}\left[z(t) = i\right] - \mu_i^*| < \delta_1.$$

Proof. Let μ^ϵ be the stationary distribution for P^ϵ. According to Theorem 3.2.1,

$$\limsup_{t=0,1,2,...} |\mathbf{Pr}\left[z(t) = i\right] - \mu^\epsilon|$$

can be made arbitrarily small with diminishing δ_2. Likewise, $|\mu^\epsilon - \mu^*|$ can be made arbitrarily small with diminishing ϵ. Combining these two bounds proves the desired result. □

Corollary 3.2.1 implies that if a state z is not a stochastically stable state then small perturbations limit the long run probability to visit this state.

Now consider stochastic stability in a "drifting" process. Stochastically stability in the process cannot be discussed in terms of a single stationary distribution because the dynamic process is non-stationary. It was proven that the resulting dynamical system can "track" drifting stochastically stable states if the drifting speed is sufficiently slow.

Theorem 3.2.2 (Lim & Shamma, 2013 [35]). *Let $\Theta \subset \mathbb{R}^m$ be a compact set. For each $\theta \in \Theta$, let $\{P_\theta^\epsilon\}$ be a regular perturbation of P_θ^0, and let μ_θ^* be the associated distribution characterizing stochastic stability. Furthermore, for each ϵ, let $\theta \mapsto P_\theta^\epsilon$ be continuous. Let the dynamic process (3) satisfy*

- $P(t) \in \left\{ P_\theta^\epsilon \;\middle|\; \theta \in \Theta \right\}$, *and*
- $\|P(t+1) - P(t)\| < \delta_2$.

For any $\delta_1 > 0$*, there exist* $\epsilon > 0$ *and* $\delta_2 > 0$ *such that*

$$\limsup_{t=0,1,2,\ldots} \left| \mathbf{Pr}\left[z(t) = i\right] - \mu^*_{\theta(t)} \right| \leq \delta_1.$$

Proof. As in the proof of Theorem 3.2.1, let the initial condition $z(0)$ be selected according to the probability distribution $x(0) \in \Delta(n)$, define the switching linear system

$$x(t+1) = P^{\mathrm{T}}(t)x(t),$$

and write

$$x(t) = \mu^*_{\theta(t)} + Ww(t).$$

Then

$$\mu^*_{\theta(t+1)} + Ww(t+1) = (P^\epsilon_{\theta(t)})^{\mathrm{T}}(\mu^*_{\theta(t)} + Ww(t)).$$

Rewriting the above equation leads to

$$\begin{aligned}
w(t+1) =& W^{\mathrm{T}}(P^\epsilon_{\theta(t)})^{\mathrm{T}}Ww(t) + W^{\mathrm{T}}(\mu^\epsilon_{\theta(t)} - \mu^*_{\theta(t+1)}) \\
&+ W^{\mathrm{T}}(P^\epsilon_{\theta(t)})^{\mathrm{T}}(\mu^*_{\theta(t)} - \mu^\epsilon_{\theta(t)}), \\
=& W^{\mathrm{T}}(P^\epsilon_{\theta(t)})^{\mathrm{T}}Ww(t) + W^{\mathrm{T}}(\mu^\epsilon_{\theta(t+1)} - \mu^*_{\theta(t+1)}) \\
&+ W^{\mathrm{T}}(\mu^\epsilon_{\theta(t)} - \mu^\epsilon_{\theta(t+1)}) \\
&+ W^{\mathrm{T}}(P^\epsilon_{\theta(t)})^{\mathrm{T}}(\mu^*_{\theta(t)} - \mu^\epsilon_{\theta(t)}),
\end{aligned}$$

where $\mu^\epsilon_{\theta(t)}$ is the stationary distribution of $P^\epsilon_{\theta(t)}$. From continuity of $\theta \mapsto P^\epsilon_\theta$, the difference between stationary distributions $\left|\mu^\epsilon_{\theta(t)} - \mu^\epsilon_{\theta(t+1)}\right|$ can be made arbitrarily small with diminishing δ_2 using continuity of stationary distributions with respect to transition probabilities (see [40]). Likewise, $\left|\mu^\epsilon_{\theta(t)} - \mu^*_{\theta(t)}\right|$ can be made arbitrarily small with diminishing ϵ. Let $A(\theta(t)) = W^{\mathrm{T}}(P^\epsilon_{\theta(t)})^{\mathrm{T}}W$. The resulting analysis resembles a slowly varying Linear Parameter Varying (LPV) system (e.g., [41, 42, 43]) of the form

$$w(t+1) = A(\theta(t))w(t) + f(t)$$

where each $A(\theta(t))$ is a stability matrix and $|f(t)|$ is arbitrarily small. Standard arguments regarding slowly varying systems lead to the desired result. □

3.3 Specialization: Game Theoretic Learning

The analysis in the previous section is in terms of Markov processes, therefore it is applicable to a variety of game theoretic learning rules. This section will apply robustness results to a specific learning rule, binary log-linear learning [12], which is a variant of standard log-linear learning [11].

Before explaining binary log-linear learning, we introduce the definition of constrained action sets first. For many practical applications, a set of available actions for a player at time t can be restricted by the player's action at time $(t-1)$. For example, suppose that an action a_i is a mobile robot's position. Then mobility constraints, such as velocity limitations or physical obstacles, imply that the set of feasible position $a_i(t+1)$ at time $(t+1)$ is constrained by its current position $a_i(t)$. Define the set valued maps for every $\mathcal{P}^i \in \mathcal{P}$ that characterize constrained action set by

$$C_i : \mathcal{A}_i \to 2^{\mathcal{A}_i}.$$

Then at each stage $t = 0, 1, 2, ...,$

$$a_i(t) \in C_i(a_i(t-1)).$$

The following characteristics are assumed for constrained action sets.

- **Feasibility :** For any player $\mathcal{P}^i \in \mathcal{P}$ and any action pair $a_i(0), a_i(m) \in \mathcal{A}_i$, there exists a sequence of actions $a_i(0) \to a_i(1) \to \cdots \to a_i(m)$ satisfying $a_i(k) \in C_i(a_i(k-1))$ for all $k \in 1, 2, \cdots, m$.
- **Reversibility :** For any player $\mathcal{P}^i \in \mathcal{P}$ and any action pair $a_i', a_i'' \in \mathcal{A}_i$, $a_i' \in C_i(a_i'') \Leftrightarrow a_i'' \in C_i(a_i')$.

Standard log-linear learning does not guarantee convergence to a potential function maximizer when players' action sets are constrained. However binary log-linear learning solves this issue. Binary log-linear learning adapts players' action as follows. At time t, one player $\mathcal{P}^i \in \mathcal{P}$ is randomly selected and all other players should repeat their current actions. The selected player $\mathcal{P}^i$ chooses one trial action $\hat{a}_i$ from its currently available action set $C_i(a_i(t-1))$ uniformly. Then the player updates its action according to the following strategy $p \in \Delta(|\mathcal{A}_i|)$.

$$p_i^{a_i(t-1)}(t) = \frac{e^{\frac{1}{\tau}U_i(a(t-1))}}{e^{\frac{1}{\tau}U_i(a(t-1))} + e^{\frac{1}{\tau}U_i(\hat{a}_i, a_{-i}(t-1))}} \tag{4a}$$

$$p_i^{\hat{a}_i}(t) = \frac{e^{\frac{1}{\tau}U_i(\hat{a}_i, a_{-i}(t-1))}}{e^{\frac{1}{\tau}U_i(a(t-1))} + e^{\frac{1}{\tau}U_i(\hat{a}_i, a_{-i}(t-1))}} \tag{4b}$$

$$p_i^{a_i}(t) = 0, \qquad \forall a_i \neq a_i(t-1), \hat{a}_i. \tag{4c}$$

The player uses a Boltzmann distribution to randomly select between its previous action $a_i(t-1)$ and randomly selected trial action $\hat{a}_i$. It was proved by Marden and Shamma that potential function maximizers being stochastically stable in binary log-linear learning.

Proposition 3.3.1 (Marden & Shamma, 2012 [12])**.** *Let G be a finite n-player potential game. Binary log-linear learning with perturbation parameter $\epsilon = e^{-1/\tau}$ defines a regular perturbation of an associated limiting Markov chain. Let μ^* be the associated distribution characterizing stochastic stability. Then*

$$\mu^*(a) \neq 0 \Leftrightarrow a \in \mathcal{A}^*.$$

Now our analysis about stochastic stability is applied to the learning rules. Consider parameter dependent utility functions

$$U_i : \mathcal{A} \times \Theta \to \mathcal{R}$$

where $\Theta \subset \mathcal{R}^m$ is compact. Assume that mapping $\theta \mapsto U_i(a;\theta)$ is continuous for all $a \in \mathcal{A}$. Then parameter dependent versions of binary log-linear rules (4) can be defined. The following two corollaries are direct consequences of the analysis in the previous section.

Corollary 3.3.1 (Lim & Shamma, 2013 [35]). *Suppose $\theta_o \in \Theta$ results in a potential game. Let $\mu^{\tau}_{\theta_o}$ be the stationary distribution of either log-linear learning or binary log-linear learning. Then for every $\delta_1 > 0$, there exists a $\delta_2 > 0$ such that*

$$|\theta(t) - \theta_o| < \delta_2$$

for all $t = 0, 1, 2, ...$ implies that

$$\limsup_{t=0,1,2,...} |\mathbf{Pr}\left[a(t) = a\right] - \mu^{\tau}_a| < \delta_1$$

for all $a \in \mathcal{A}$.

Corollary 3.3.2 (Lim & Shamma, 2013 [35]). *Suppose that any $\theta \in \Theta$ results in a potential game with potential function $\phi(a; \theta)$. Then for either log-linear learning or binary log-linear learning, for every $\delta_1 > 0$, there exists a $\delta_2 > 0$ and $\tau > 0$ such that*

$$|\theta(t+1) - \theta(t)| < \delta_2$$

for all $t = 0, 1, 2, ...$ implies that

$$\liminf_{t=0,1,2,...} \mathbf{Pr}\left[\phi(a(t); \theta(t)) = \max_{a \in \mathcal{A}} \phi(a; \theta(t))\right] > 1 - \delta_1.$$

An implication of Corollary 3.3.2 is that the joint action $a(t)$ plays the potential function maximizer for current parameter $\theta(t)$ with high probability. Therefore $a(t)$ "tracks" the stochastically stable state.

So far, it was assumed that all players share the same values of $\theta(t)$. Now assume that each player independently observes $\theta(t)$ with some bounded noise. Define $\hat{\theta}^i(t)$ as an observed $\theta(t)$ by player i. Then the learning rule (4) is modified as follows.

$$\begin{aligned}
p_i^{a_i(t-1)}(t) &= \frac{e^{\frac{1}{\tau}U_i(a(t-1);\hat{\theta}^i(t))}}{e^{\frac{1}{\tau}U_i(a(t-1);\hat{\theta}^i(t))} + e^{\frac{1}{\tau}U_i(\hat{a}_i,a_{-i}(t-1);\hat{\theta}^i(t))}} \\
p_i^{\hat{a}_i}(t) &= \frac{e^{\frac{1}{\tau}U_i(\hat{a}_i,a_{-i}(t-1);\hat{\theta}^i(t))}}{e^{\frac{1}{\tau}U_i(a(t-1);\hat{\theta}^i(t))} + e^{\frac{1}{\tau}U_i(\hat{a}_i,a_{-i}(t-1);\hat{\theta}^i(t))}}. \\
p_i^{a_i}(t) &= 0, \qquad \forall a_i \neq a_i(t-1), \hat{a}_i
\end{aligned}$$

Note that event density $\theta(t)$ is replaced to its noisy measurement $\hat{\theta}^i(t)$ from (4). The following corollary can be easily derived from Corollary 3.3.1 and Corollary 3.3.2.

Corollary 3.3.3 (Lim & Shamma, 2013 [35]). *Suppose that any $\theta \in \Theta$ results in a potential game with potential function $\phi(a, \theta)$. Then for either log-linear learning or binary log-linear learning, for every $\delta_1 > 0$, there exist $\delta_2 > 0$, $\delta_3 > 0$ and $\tau > 0$ such that*

$$|\theta(t+1) - \theta(t)| < \delta_2$$

$$\left|\hat{\theta}^i(t) - \theta(t)\right| < \delta_3$$

for all $t = 0, 1, 2, ...$ implies that

$$\liminf_{t=0,1,2,...} \mathbf{Pr}\left[\phi(a(t); \theta(t)) = \max_{a \in \mathcal{A}} \phi(a; \theta(t))\right] > 1 - \delta_1.$$

3.4 Example : Sensor coverage problem

As an illustrative example, this section presents a dynamic sensor coverage problem. Consider a group of n mobile robots with limited sensor range trying to cover a finite mission space S. Each robot $i \in \{1, 2, ..., n\}$ optimizes its position over S therefore action space $\mathcal{A}_i = S$ for all i. Available actions at time $(t+1)$ for a robot i are its current section $a_i(t)$ and adjacent sections including diagonal directions. Let $\theta(t)$ be a time variant event density function $\theta(t) \in \Delta(|S|)$ over the mission space S. It was assumed that $|\theta(t+1) - \theta(t)|$ was always bounded by some $\delta_2 \geq 0$ for all t.

The global objective $\phi(a; \theta(t))$ at time t is a sum of event density over sections that is covered by mobile robots, i.e.

$$\phi(a; \theta(t)) = \sum_{s_j \in S} \mathbf{I}_a(s_j)\theta_j(t) \tag{5}$$

where $\mathbf{I}_a : S \to \{0, 1\}$ is an indicator function defined as $\mathbf{I}_a(s_j) = 1$ if s_j is in the range of at least one sensor under the joint action a, and $\mathbf{I}_a(s_j) = 0$ otherwise. Then define player's utility function as its marginal contribution to a global objective, i.e.

$$U_i(a_i, a_{-i}; \theta(t)) = \phi(a_i, a_{-i}; \theta(t)) - \phi(a_i^0, a_{-i}; \theta(t)) \tag{6}$$

where a_i^0 is a null action which is equivalent to player i turning off its sensor. The utility structure in (6) is called Wonderful Life Utility (WLU) [22], and it was proved that WLU

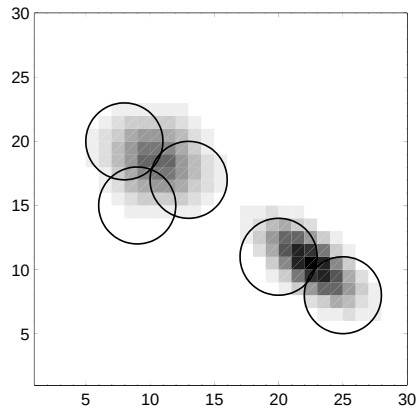


Figure 3: Example of sensor allocation and event density over the mission space

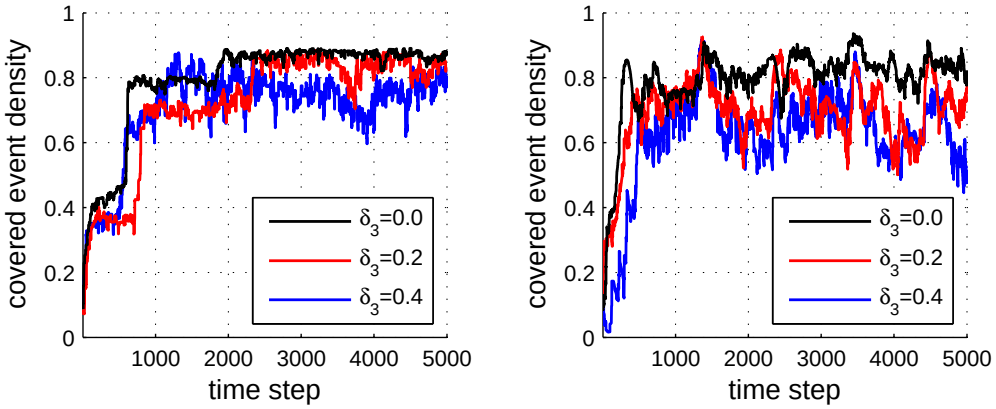
constitutes a potential game [8] with the global objective ϕ as a potential function of the game.

Mobile robots adopt binary log-linear learning with noisy theta measurement $\hat{\theta}_i$ from the previous section. It was assumed that a size of measurement noise $|\theta(t) - \hat{\theta}_i(t)|$ is bounded by δ_3. Figure 3 shows an example of optimal sensor allocation and event density over the mission space. Each circle in the figure indicates coverage of a sensor.

3.4.1 Simulations

In the simulations, five mobile robots ($n = 5$) played binary log-linear learning over the mission space $S = \{1, 2, ..., 30\} \times \{1, 2, ..., 30\}$. Each robot has a finite sensor range of 3 unit length. The temperature parameter τ for learning algorithm was 0.005. Both static and dynamic event densities were simulated with various noise levels.

In Fig. 4 (a), the event density distribution $\theta(t)$ was time-invariant. Since $\theta(t) = \theta(0)$ for all t, $|\theta(t+1) - \theta(t)|$ was always bounded by $\delta_2 = 0$. The three cases when a size of randomly generated noise, $|\hat{\theta}^i(t) - \theta(t)|$, was bounded by $\delta_3 = 0.0, 0.2$ or 0.4 were tested. As stated in Corollary 3.3.3, a joint strategy converged to a neighborhood of the potential maximizer even under the presence of noise, and smaller δ_3 yields more stable behavior. In all cases, the players were maintaining the coverage successfully after the players they to neighborhood of an optimal setting.



(a) Static event (b) Dynamic event

Figure 4: Event coverage simulation results (a) static events (b) dynamically drifting events. A size of randomly generated noise δ_3 was bounded by 0.0,0.2, or 0.4 for both cases.

Now consider the case when event density is time variant. Fig. 4 (b) displays simulation results with time-variant event density. In the simulations, the two centers of events moved with a limited speed less than 0.03 per time step, therefore $|\theta(t+1) - \theta(t)|$ was bounded for all t. The results of time-variant event coverage were less stable than the time-invariant cases, but still a group of players could follow and cover the events. Although the results of dynamic event cases were less stable than static event cases, the robots were able to follow the moving events. The moving trajectories of drifting events and mobile robots from $t = 3500$ to $t = 4000$ are illustrated in Figure 5. In the figure, the five mobile robots could cover dynamically drifting events very well.

3.4.2 Laboratory experiments

3.4.2.1 Experimental system

A heuristic variant of log-linear learning was tested in an experimental setup with Khepera III two-wheeled mobile robots [44] and OptiTrack motion capture system [45]. Figure 6 and Figure 7[2] shows the Khepera III robot and OptiTrack motion capture camera respectively.

Khepera III is a mobile robot model developed by K-team corporation and the robot

[2]The image is taken from [45]

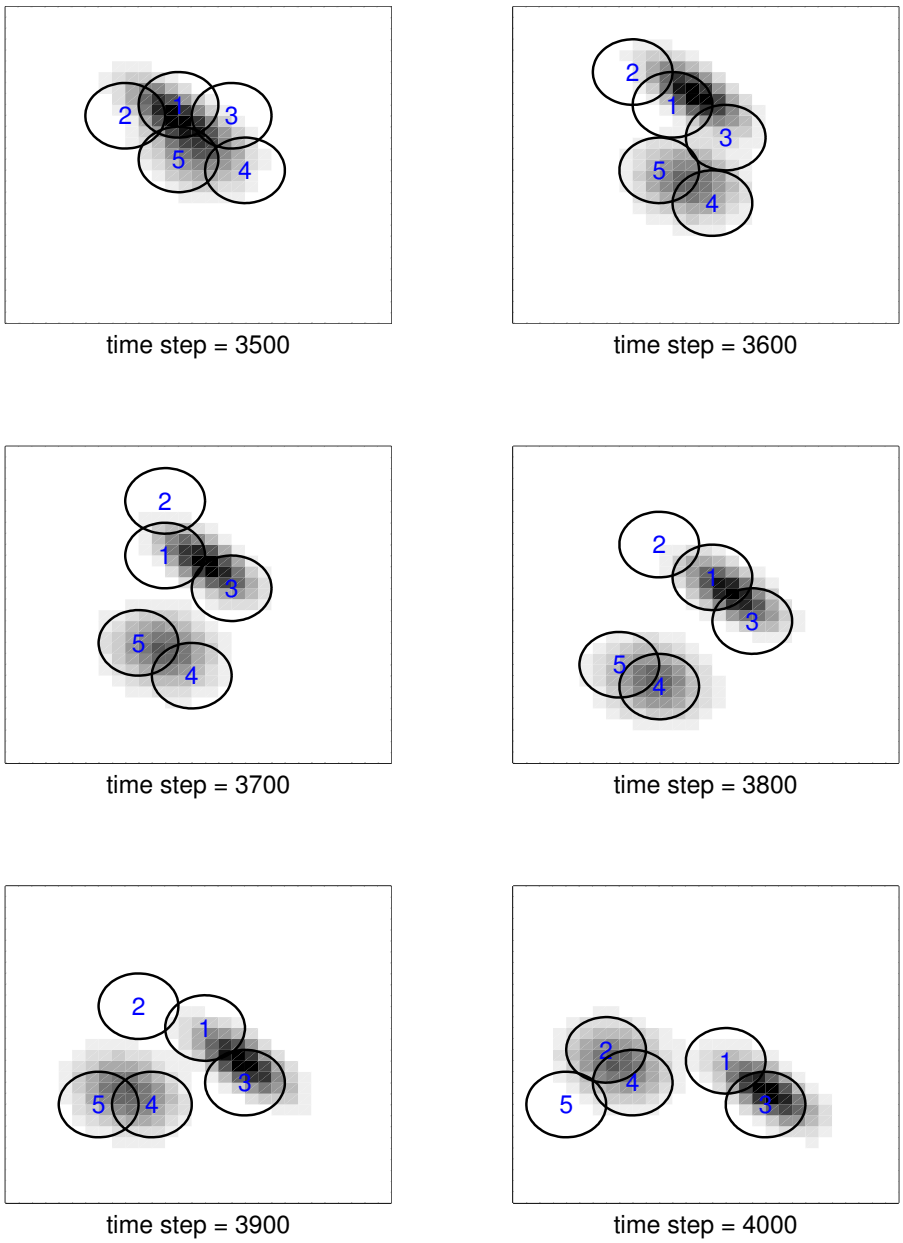


Figure 5: Snapshots of a dynamic event coverage simulation at every 100 time steps from $t = 3500$ to $t = 4000$.

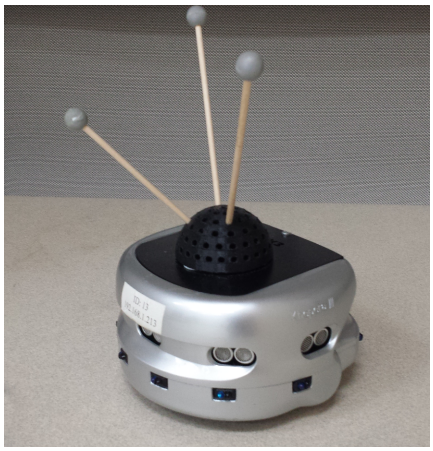


Figure 6: The Khepera III robot. The three small reflective balls are attached to the robot for 3-dimensional position tracking by the OptiTrack system.

base uses the KoreBot computer board which is develepoed by the same manufacturer. The size of each Khepera III is 130mm $\times$ 70mm and its maximum running speed is 0.5m/s. The KoreBot board features an embedded Linux operating system and standard compact flash extension cards supporting WiFi and Bluetooth wireless communication. For experiments, a 802.11b wireless network extension card is used, and an application program was developed using the application programming interface (API) provided by the manufacturer.

The OptiTrack motion capture system which is developed by NaturalPoint corporation was used as a global positioning system (GPS) for mobile robots. The GPS lab setting using motion tracking system is based on [46, 47]. Using the features of OptiTrack motion cation capture system, real-time positions of the mobile robots are tracked and recorded by a host computer of the experiment system.

The entire experiment system structure is demonstrated in Figure 8. A virtual event density $\theta(t)$ over the mission space was generated by a host computer. As an indoor Global Positioning System (GPS) for mobile robots, the motion capture system transmits to a robot its current position and event density $\theta_j(t)$ for all section s_j which is under coverage of robot i. By this experimental setup, each mobile robot could adjust its position using log-linear learning to maximize event coverage.

Figure 7: The OptiTrack motion capture camera.

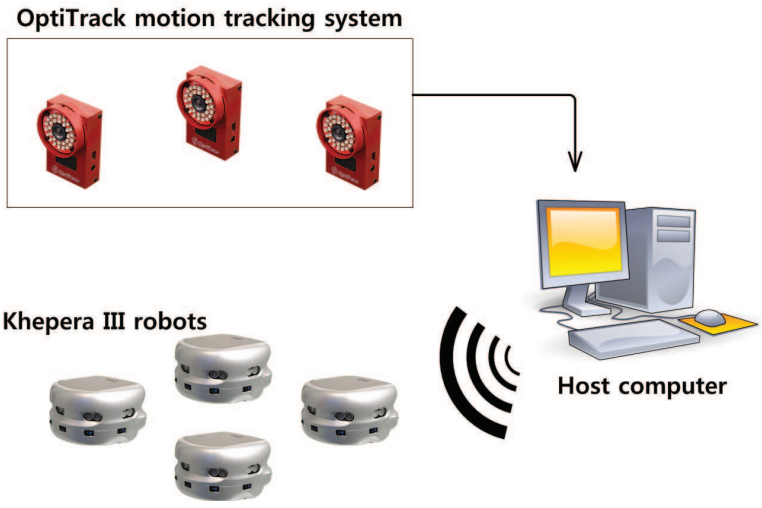
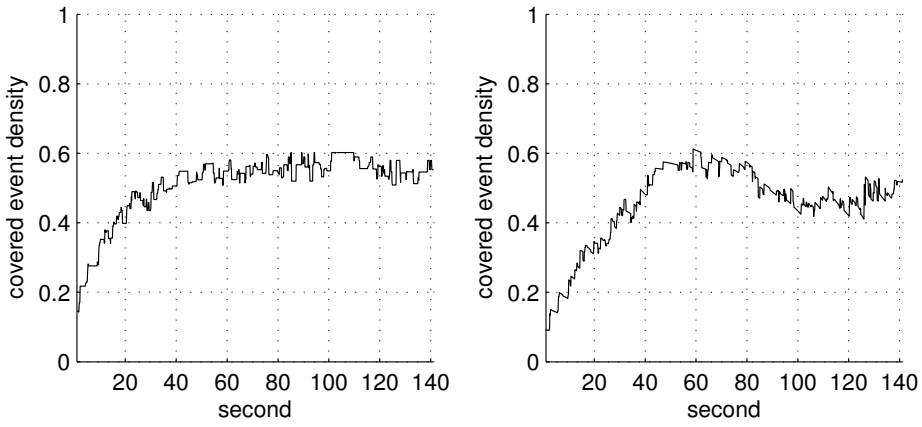


Figure 8: The structure of the experiment system



(a) Static event (b) Dynamic event

Figure 9: Event coverage experiment results (a) static events (b) dynamically drifting events

3.4.2.2 Experiment results

For experiments, four robots ($n = 4$) were used, and the missions space S was discretized into rectangular grid $\{-9, -8, ..., 0, ..., 8, 9\} \times \{-9, -8, ..., 0, ..., 8, 9\}$. Each section $s_j \in S$ was 0.2m $\times$ 0.2m, and each robot had a sensor coverage of radius 0.4m. The average moving speed of the robots was 0.075m/s. For dynamic event experiments, the moving speed of the center of event was 0.03m/s.

In the implemented variant of log-linear learning, a robot decided to stay at its current section for two seconds with probability $(1-\omega) = 0.3$, or to select new action among $C_i(a_i)$ based on its strategy with probability $\omega = 0.7$. Robots used the temperature parameter $\tau = 0.0046$ for the strategy. This heuristic variant was to include an additional incentive for exploration.

Figure 9 presents experimental results for static and dynamic events. These results indicate that robots could track the drifting distribution and the performance was comparable to the static event case. During the experiment, there was no collision between robots even though we did not consider any collision avoidance algorithm. The reason is that every robot tried to minimize overlapped coverage region to maximize its utility which was its marginal contribution to event coverage. Therefore players naturally avoided to be too

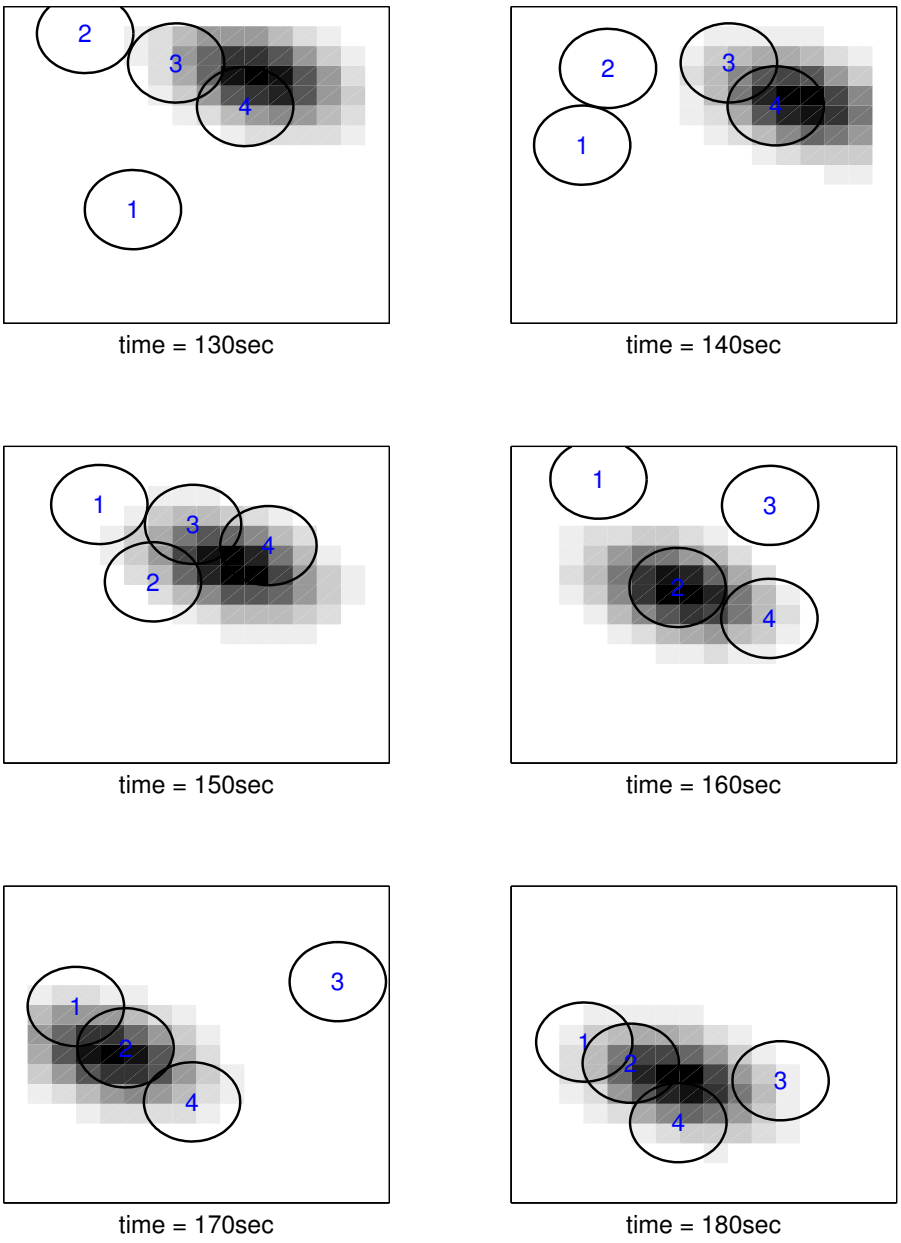


Figure 10: Snapshots of a dynamic event coverage experiment at every 10 seconds from 130 seconds to 180 seconds

close to each other. The moving trajectories of drifting events and mobile robots from 130 seconds to 180 seconds are shown in Figure 10.

3.5 Summary

We have examined the robustness of stochastic stability with unstructured noise or slowly drifting environments. While the results were specialized to log-linear learning, the methods were derived for Markov chains and hence can be valid for a variety of settings. The approach was applied in both simulation and laboratory experiments to distributed area coverage with mobile robots.

CHAPTER 4

COORDINATING WITHOUT COMMUNICATIONS

4.1 Introduction

This chapter considers a problem of coordinating players without communications. The specific model considered in this research is a team-based zero-sum game where a team of agents wants to maximize the team's reward against hostile opponents. The agents within the team cannot communicate with each other, so they should independently optimize their actions using local information. We will assume that players within the team do not know the payoff structure of the game and cannot observe other players' actions. Hence a learning rule for the team players should be a payoff-based rule.[1] There are two central concerns in this work: i) learning correlated behavior with payoff-based dynamics and ii) rationality level of opponents.

In general team-based zero-sum games, the team is not regarded as a single player even though the team players have identical payoffs because the team members might not be able to coordinate their actions. A team-maxmin equilibrium [48] is an equilibrium of team-based zero-sum games in which coordination is not available. Given inability to coordinate team players actions, the team-maxmin equilibria are equilibria of the game with the highest payoff to the team. A team-maxmin strategy may not be unique but the payoff of the team-maxmin equilibrium is unique.

If coordination between team members is available, the team's payoff can be improved beyond the team-maxmin equilibria. Coordination of players requires observation of a common signal generated from an external device [49] or communication between players [50]. A correlated equilibrium [49] which is a generalization of a Nash equilibrium is an equilibrium of the game when full coordination is available. In a team-based zero-sum game, opponents may be also able to observe and take advantage of information in

[1] For more information about payoff-based dynamics, refer to Chapter 2.

the common signal or communications. The tradeoffs for the team between generation of signals for future correlation and use of correlation for present payoffs is studied in [51].

Achieving correlated behavior without common signals or communication have been studied by many works. One of the promising class of such learning rules is based on the concept of regret such as [2, 52, 53, 54, 55]. Specifically, a recently discovered payoff-based learning procedure of [56] is closely related to this research. In [56], players can learn series of actions whose empirical distribution of play shows correlated behavior with no explicit communications. The idea of optimizing a periodic action sequence instead of adapting one-shot strategy is also adopted in our research. However the method described in [56] does not assume the presence of opponents. Hostile opponents in a team-based zero-sum game make learning correlated behavior trickier.

The opponent player in our team-based zero-sum game setting can observe the team players actions and consequently he has more information than the team players. If the opponent has perfect rationality and infinite memory, there always exist perfect strategies for the opponent against periodic pure action sequences of the team players. Now assume that the opponent player's rationality is limited. Can the team players deceive the opponent in spite of information asymmetry? To answer the question, a mathematical model of the opponent's rationality should be defined.

Behavior of players with limited rationality in repeated games has been studied in the game theory literature. There is a rich set of research like [57, 58, 59, 60, 61] that adopts the notion of finite automata to define the bounded rationality of agents. We adopt a bounded recall strategy [62], which is a special class of automata, as the opponent's rationality model.

The contribution of this work is to demonstrate a payoff-based correlated behavior learning under the presence of a hostile opponent. If the opponent has a specific form of bounded rationality, then the team players' periodic and deterministic action sequences can achieve performance close to the correlated equilibrium. Since coordinating periodic

and deterministic action sequences does not required common randomness, team players can learn their action sequences without communications or public signals.

One of the problems with an action sequence learning is that the size of the game grows exponentially as the length of the action sequence gets longer. We introduce a notion of micro-players for an efficient sequence learning. By using de Bruijn sequences with the micro-player approach, the size of game can be independent from the sequence length.

The outline of this chapter is as follows. In Section 4.2, some important preliminary concepts related to this research are introduced. Next, in Section 4.3, a team-based zero-sum game is illustrated as a motivational example. In Section 4.4, a mathematical model of an opponent player with limited rationality is defined, and we prove that there exist optimal periodic sequences that look random from the opponent player's point of view. Section 4.5 introduces a notion of micro-players to learn a long action sequence efficiently with no communications. Then two types of learning methods based on the micro-players are proposed in Section 4.6 and Section 4.7. Lastly, in Section 4.8, we conclude with a summary of the chapter and discussion about some limitations of the proposed methods.

4.2 Preliminaries

4.2.1 Correlated equilibrium

The idea of correlated equilibria is that each player correlates its strategy to the same public signal by assigning an action to every possible signal observation. If no player wants to unilaterally deviate from its current strategy, the joint strategy is a correlated equilibrium.

Definition 4.2.1 (Aumann, 1974[49]). *A joint strategy $\mu \in \Delta(A)$ is a correlated equilibrium if for $\forall i \in N$, $\forall a_i \in A_i$*

$$\sum_{(a_i, a_{-i}) \in A} \mu_{(a_i, a_{-i})}(u_i(a_i, a_{-i}) - u_i(a'_i, a_{-i})) \geq 0$$

where $\mu_{(a_i, a_{-i})}$ is a probability of a joint action (a_i, a_{-i}) in the joint strategy μ.

Correlated equilibria have some useful properties. First, correlated equilibria can be found in polynomial time for a succinct representation of a game [63]. And any convex

combination of correlated equilibria is also a correlated equilibrium. Moreover, every Nash equilibrium has a corresponding correlated equilibrium. Note that not every correlated equilibrium is a Nash equilibrium, therefore a correlated equilibrium is a weaker notion than a Nash equilibrium.

4.2.2 De Bruijn sequences

The definition of De Bruijn sequences [64] is as follows.

Definition 4.2.2 (De Bruijn, 1946[64])**.** *A De Bruijn sequence is a cyclic sequences of a given alphabet set A with size α for which every possible subsequence of length β in A appears as a sequence of consecutive characters exactly once.*

Let $\mathcal{B}(\alpha, \beta)$ be the set of all De Bruijn sequences of order β from a symbol set A with size α. For an example, one of possible De Bruijn sequences in $\mathcal{B}(2, 3)$ for a symbol set $A = \{0, 1\}$ is (00010111). Each De Bruijn sequence in $\mathcal{B}(\alpha, \beta)$ has length α^β. For all α and β, the size of the set of De Bruijn sequences is as follows.

$$|\mathcal{B}(\alpha, \beta)| = \frac{(k!)^{\alpha^{\beta-1}}}{\alpha^\beta}$$

A β-dimensional De Bruijn graph over α symbols is a directed graph whose vertices are all available β-length tuples over α symbols. If one of the tuples can be built from another tuple by shifting all its symbols by one place to the left and adding a new symbol at the end, then the former tuple has a directed edge from the latter. Figure 11 shows a 3-dimentional De Bruijn graph over a symbol set $\{0, 1\}$.

Every de Bruijn sequence $\mathcal{B}(\alpha, \beta)$ corresponds to a Eulerian cycle of a $(\beta-1)$-dimensional De Bruijn graph or a Hamiltonian cycle of β-dimensional De Bruijn graph [65]. For an example, consider a sequence (00010111) which is a De Bruijn sequence $\mathcal{B}(2, 3)$ for $A = \{0, 1\}$ again. Every tuple with length three occurs exactly once if every vertex of Figure 11 is visited exactly once. Therefore one can build a De Bruijn sequence in $\mathcal{B}(\alpha, \beta)$ with arbitrary α and β using any algorithms finding Eulerian cycle over a given graph.

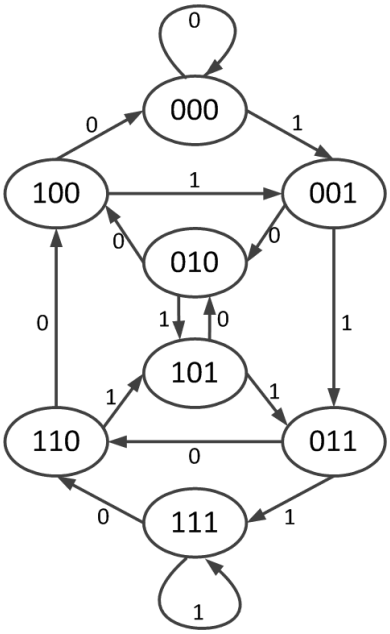


Figure 11: A 3-dimentional De Bruijn graph over a symbol set $\{0, 1\}$. Every Eulerian cycle of the graph is an element of the de Bruijn sequence set $\mathcal{B}(2, 3)$.

In our work, a De Bruijn sequence is given to players during an initialization stage before starting a game. For any arbitrary α and β, a de Bruijn sequence $\mathcal{B}(\alpha, \beta)$ can be constructed using established methods such as shifted registers [66, 67] or finite fields [68]. Discovering and analysing better methods to construct a de Bruijn sequence is an ongoing research topic in combinatorics and cryptography [69, 70]. To create De Bruijn sequences, one can utilize any methods including algorithms that are not listed here, and we will not explicitly discuss how the sequence is generated in this thesis.

4.2.3 Information theory tools [2]

Let X be a discrete random variable with a probability distribution $p \in \Delta(|X|)$. The entropy $H(X)$ is defined by

$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$

where $p(x)$ is a probability of x in the distribution p. The entropy is a measure of uncertainty in the random variable. By the convention, $0 \log_2 0 = 0$.

[2]The contents on this subsection refers to [71].

Let (X, Y) be a couple of discrete random variables with a joint probability distribution $p \in \Delta(|X \times Y|)$. The conditional entropy $H(X|Y)$ is

$$\begin{aligned} H(X|Y) &= \sum_y p(y) \sum_x p(x|y) \log_2 p(x|y) \\ &= \sum_y p(y) H(X|Y = y). \end{aligned}$$

And the following chain rule of entropy can be easily verified.

$$\begin{aligned} H(X, Y) &= H(X) + H(Y|X) \\ &= H(Y) + H(X|Y) \end{aligned}$$

The mutual information of two discrete random variables X and Y is

$$I(X; Y) = H(X) + H(Y) - H(X, Y).$$

The mutual information $I(X; Y)$ means the reduction in the uncertainty of X due to the knowledge of Y.

In analogy to conditional entropy, conditional mutual information can be defined. For discrete random variables X, Y, and Z, the mutual information of X and Y given Z is defined by

$$I(X; Y|Z) = H(X|Z) + H(Y|Z) - H(X, Y|Z).$$

4.3 Team-based zero-sum game

Consider a team based zero-sum game $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ where $\phi : X_1 \times X_2 \times Y \to \mathbb{R}$ is a reward for a team $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$. If a joint strategy of the team $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$ and the opponent player $\mathcal{P}_Y$ are given as μ and σ respectively, we will allow abuse of notation by writing $\phi(\mu, \sigma)$ as an average reward for the team. Denote actions for $\mathcal{P}_X^1$, $\mathcal{P}_X^2$ and $\mathcal{P}_Y$ as $x^1 \in X_1$, $x^2 \in X_2$, and $y \in Y$ respectively. Suppose that the action set for each player is $X_1 = X_2 = Y = \{1, 2\}$. The team $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$ wants to maximize their reward ϕ and the opponent $\mathcal{P}_Y$ wants to minimize it. The payoff $\phi(x^1, x^2, y)$ to the team are given by

	1	**2**
1	1	0
2	0	0

$y =$**1**

	1	**2**
1	0	0
2	0	1

$y =$**2**

where $\mathcal{P}_X^1$ chooses rows, $\mathcal{P}_X^2$ chooses columns, and $\mathcal{P}_Y$ chooses matrices. If three players' actions are the same then a reward for the team is 1. Otherwise the team gets 0 point.

If $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ independently mix their strategies, i.e., strategies of $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ are not correlated, then the maxmin payoff for the team is $\frac{1}{4}$ and a maxmin joint strategy $\mu^*_{\text{ind}} \in \Delta(X_1) \times \Delta(X_2)$ is

$$\mu^*_{\text{ind}} = \underset{\mu \in \Delta(X_1)\times\Delta(X_2)}{\arg\max} \left(\min_{\sigma \in \Delta(Y)} \phi(\mu, \sigma) \right) = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix}.$$

Now assume that there exist some secret signals between $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ so that full correlation is achievable. Then the maxmin payoff is $\frac{1}{2}$ and a maxmin joint strategy $\mu^*_{\text{cor}} \in \Delta(X_1 \times X_2)$ is as follows.

$$\mu^*_{\text{cor}} = \underset{\mu \in \Delta(X_1\times X_2)}{\arg\max} \left(\min_{\sigma \in \Delta(Y)} \phi(\mu, \sigma) \right) = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

In the cases when full correlation is available, the team can be considered as one player who selects its action from $(X_1 \times X_2)$. However correlation tools such as communication or public signals are not always available in many problem settings. Then how can the group of players converge to a correlated equilibrium if any correlation tool is not available?

The main idea of this research is to build deterministic and periodic action sequences that look random to opponents with limited rationality. Suppose that $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ play a deterministic and cyclic joint action sequence whose empirical frequency of a joint action is equivalent to a correlated equilibrium. If there is no restriction on the opponent's strategic capability, it is obvious that an opponent can always win against the deterministic players. However agents in many practical applications have bounded rationality, and even a

computer can be considered as boundedly rational because of its limited memory size or processor speed. If the opponent player $\mathcal{P}_Y$ cannot recognize cyclic patterns longer than a certain length, then a sufficiently long cyclic joint action sequence of the team may perform as well as a sequence of i.i.d. random joint actions.

4.4 Opponents with limited rationality

This section defines bounded rationality of the opponent player and proves existence of optimal periodic action sequences against the opponent. For this section only, we will consider a team of players $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$ as one player $\mathcal{P}_X$. Then the game illustrated in the previous section becomes a repeated two player zero-sum game $\mathbf{G} = (\{\mathcal{P}_X, \mathcal{P}_Y\}, \{X, Y\}, \phi)$ where $\phi : X \times Y \to \mathbb{R}$ is a reward for $\mathcal{P}_X$. Suppose that $\mathcal{P}_X$ plays a L-periodic pure action sequence $x \in (X)_L \subset X^{\mathbb{N}}$ where $(X)_L$ is the set of all L-periodic sequences from X. Assume that $\mathcal{P}_Y$ plays a pure k-recall strategy $\sigma : \bigcup_{n=0}^{\infty} X^n \to Y$ as follows.

Definition 4.4.1 (Lehrer, 1988 [62])**.** *A pure k-recall strategy for agent $\mathcal{P}_Y$ is a function $\sigma : \bigcup_{n=0}^{\infty} X^n \to Y$ satisfying $\forall n > k$, $\forall x \in X^n$*

$$\sigma(x) = \sigma(x_{n-k+1}, ..., x_n)$$

Let Σ_k be the set of all pure k-recall strategies for $\mathcal{P}_Y$ and we will assume that $k < L$. Note that a player with k-time step memory is able to play only bounded recall strategies of order k or less.

For a given $\sigma \in \Sigma_k$, define $p_\sigma^* \in \Delta(X)$ as the optimal strategy of $\mathcal{P}_X$ that maximizes an average reward when $\mathcal{P}_X$ plays i.i.d. random actions, i.e.,

$$p_\sigma^* = \arg\max_{p \in \Delta(X)} \phi(p, \sigma).$$

Then we want to find an action sequence $x \in (X)_L$ that looks i.i.d. random with distribution $p^* \in \Delta(|X|)$ to observers with k or less memory.

Define $\hat{x}_t$ as $\mathcal{P}_X$'s action observed by $\mathcal{P}_Y$ at time t, i.e., $\hat{x}_t = x_{t+\tau}$ where τ is selected uniformly random from $\{0, ..., L-1\}$. Therefore the observed sequence $(\hat{x}_1, \hat{x}_2, ...)$ is

equivalent to τ-step shifted sequence x. Even though the original sequence x is deterministic, $\hat{x}_t$ is a random variable with distribution equivalent to an empirical frequency of x. Let $\mathbf{Pr}\left[\hat{x}_t|\hat{x}_{t-k},...,\hat{x}_{t-1}\right] \in \Delta(X)$ be a probability distribution of $\hat{x}_t$ over X when $\hat{x}_{t-k},...,\hat{x}_{t-1}$ are given. Then we will prove the following theorem.

Theorem 4.4.1. *Assume that a finite set X, a real number $\epsilon > 0$ and a positive integer k are given. There exists L that guarantees existence of L-periodic sequence $x \in (X)_L$ such that for any $t > k$*

$$\|p^* - \mathbf{Pr}\left[\hat{x}_t|\hat{x}_{t-k},...,\hat{x}_{t-1}\right]\| < \epsilon.$$

It can be interpreted as that any observer with k-memory cannot benefit from its memory about $\hat{x}_{t-1},...,\hat{x}_{t-k}$ when it tries to predict $\mathcal{P}_X$'s action $\hat{x}_t$.

To prove the theorem, we will use De Bruijn sequences. If p^* is a uniform distribution over X, then De Bruijn sequences in $\mathcal{B}(|X|, k+1)$ satisfy the property stated in the theorem. For examples, consider a De Bruijn sequence '00010111' again. For a 2-memory observer, this sequence looks uniformly random over $\{0, 1\}$. Assume that the observer observed '00' during last two time steps. The empirical frequency of '000' and '001' is the same, so the memory of the last two time steps is not helpful to predict the next action is whether '0' or '1'.

Now consider the case with a more general p^* rather than uniform distribution. Define $Q_m \subset \Delta(X)$ as the set of all available empirical frequencies of m-length sequence from X. For example, if $X = \{0, 1\}$ then

$$Q_m = \left\{ \left(\frac{0}{m}, \frac{m}{m}\right), \left(\frac{1}{m}, \frac{m-1}{m}\right), ..., \left(\frac{m}{m}, \frac{0}{m}\right) \right\}.$$

Let $\hat{p}^* = \underset{p \in Q_m}{\arg\min} \|p^* - p\|$ be the closest probability from p^* in Q_m. The larger the value m is, the more precisely an arbitrary distribution p^* can be approximated by $\hat{p}^*$.

Consider a De Bruijn sequence $s^{\mathcal{B}} \in \mathcal{B}(m, k+1)$ of a set $A = \{0, 1, ..., m-1\}$. Then one can define a mapping function $\pi : A \to X$ such that for $\forall j \in X$

$$|\pi^{-1}(j)| = m\hat{p}^*_j$$

Define $x^{\mathcal{B}} \in (X)_{m^{k+1}}$ as a m^{k+1}-periodic sequence canonically induced from $s^{\mathcal{B}}$ by π.

Let $w = (w_1, w_2, ..., w_{k+1}) \in \prod_{j=1}^{k+1} X$ be a $(k+1)$-length word of X, and EF(x, w) be an empirical frequency of word w in a sequence x. Then for all $w \in X^{k+1}$,

$$\text{EF}(x^{\mathcal{B}}, w) = \prod_{i=1}^{k+1} \hat{p}^*_{w_i}.$$

Hence

$$\mathbf{Pr}\left[\hat{x}^{\mathcal{B}}_t | \hat{x}^{\mathcal{B}}_{t-1}, ..., \hat{x}^{\mathcal{B}}_{t-k}\right] = \mathbf{Pr}\left[\hat{x}^{\mathcal{B}}_t\right] = \hat{p}^*$$

where $\hat{x}^{\mathcal{B}}$ is randomly shifted sequence $x^{\mathcal{B}}$. Note that $\|p^* - \hat{p}^*\|$ can be made arbitrarily small by selecting arbitrarily large m.

As an example, let $k = 1$, $X = \{A, B\}$, and $p^* = (2/3, 1/3) \in \Delta(X)$. Let $m = 3$ then $\hat{p}^*$ is equivalent to p^* because $p^* \in Q_3$. Consider $s^{\mathcal{B}} = 001021122$ which is one of De Bruijn sequence in $\mathcal{B}(m, k+1) = \mathcal{B}(3, 2)$. Now consider a mapping function $\pi : \{0, 1, 2\} \to X$ such that $\pi(0) = \pi(1) = A$ and $\pi(2) = B$. Then $x^{\mathcal{B}}$ induced from $s^{\mathcal{B}}$ is $AAAABAABB$ and it satisfies

$$\mathbf{Pr}\left[x^{\mathcal{B}}_t | x^{\mathcal{B}}_{t-1}\right] = \mathbf{Pr}\left[x^{\mathcal{B}}_t\right] = p^*.$$

Now apply the theorem to two player zero-sum games. Define Σ_k as the set of all k-recall strategies for the second player. Then the following corollary is a direct consequence of Theorem 4.4.1.

Corollary 4.4.1. *Consider a two player zero-sum game $G = \langle \{\mathcal{P}_X, \mathcal{P}_Y\}, \{X, Y\}, \phi \rangle$. There exists a positive integer L such that, for any $\sigma \in \Sigma_k$ and $\epsilon > 0$, guarantees existence of $x \in (X)_L$ as follows.*

$$\max_{p \in \Delta(X)} \mathbb{E}(\phi(p, \sigma)) - \epsilon < \lim_{T \to \infty} \frac{1}{T} \sum_{i=k+1}^{T} \phi(x_i, \sigma(x_{i-k}, ..., x_{i-1}))$$

Therefore if $\mathcal{P}_X$ selects a period of its action sequence, L, large enough, then there exists a sequence $x \in (X)_L$ that looks almost like the optimal one-shot strategy $p^* \in \Delta(X)$ from $\mathcal{P}_Y$'s point of view. The result proves the existence of effective joint periodic action sequences against opponents with bounded rationality in team-based zero-sum games.

4.5 Micro-players within a team player

This section discusses how the team of players $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$ can coordinate periodic joint action sequences without communications, and introduces a notion of micro-players. Consider a team-based zero-sum game $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ where $\phi : X_1 \times X_2 \times Y \to \mathbb{R}$ is a reward for the team $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$. Let X be a joint action set $X_1 \times X_2$. Each $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ plays a L-periodic pure action sequence and $\mathcal{P}_Y$ plays k-recall strategy. Denote periodic action sequences of $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ as $x^1 \in (X_1)_L$ and $x^2 \in (X_2)_L$ respectively. Let x be a joint action sequence (x^1, x^2).

Assume that $\mathcal{P}_Y$ can observe all other players actions, but each $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ can only observe its own action and reward. Any form of communication is not allowed between team players, therefore $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ should optimize their action sequences using payoff-based learnings. We will assume that $\mathcal{P}_Y$'s strategy σ is static for the rest of this chapter. For a fixed opponent's strategy $\sigma \in \Sigma_k$, a team based zero-sum game G can be considered as an identical interest game $\mathbf{G}' = (\{\mathcal{P}_X^1, \mathcal{P}_X^2\}, \{(X_1)_L, (X_2)_L\}, \phi_\sigma^L)$ between $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$. The reward function $\phi_\sigma^L : (X_1)_L \times (X_2)_L \to \mathbb{R}$ is defined as

$$\phi_\sigma^L(x^1, x^2) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=k+1}^{T} \mathbb{E}\phi(x_t, \sigma(x_{t-k}, ..., x_{t-1})) \tag{7}$$

where $x_t = (x_t^1, x_t^2)$. In other words, ϕ_σ^L is the team's average reward for playing periodic sequence $x \in (X)_L$ against the opponent's strategy σ. Because identical interest games are a special class of potential games, one can use any general payoff based learning for potential games in which 'actions' are now replaced with 'action sequences'.

The next question is how to learn and coordinate the optimal sequence. A straightforward approach is handling an action sequence as an action with a higher dimension. Theoretically, there is no big difference between adapting 'action sequences' and adapting 'actions'; 'an action sequence' can be treated as 'an action with a higher dimension.' In a conventional repeated game setting, a process of general learning algorithms developed for a single action learning are as follows; i) at every time step, a player i plays a selected

action $a_i \in \mathcal{A}_i$, ii) observes the result and iii) updates its internal state. If a player wants to learn a L-length action sequence with a general learning algorithm developed for a single action learning, then the aforementioned process can be modified as follows; i) at every L-time steps, a player i plays a selected L-length action sequence $a_i \in \mathcal{A}_i^L$, ii) observes the result during the period and iii) updates its internal state at the end of the period. Hence many existing game theoretic learning algorithms could be applied without drastic changes.

However learning a sequence of actions with payoff-based approach can be painfully slower than the cases in which a player learns just a single action. In a single action learning, a number of action candidates is the size of the action set, i.e., $|\mathcal{A}_i|$, and player can test one action at a time. On the other hand, for a L-length action sequence learning, the size of the candidate sequence set, i.e., $|\mathcal{A}_i^L|$ is exponentially increased, and it takes L-time steps to test the result of the selected sequence. Generally the larger action set results in the slower adapting speed in payoff-based learning because players often depend on their 'luck' to choose a trial action. For this reasons one can easily expect that this approach might be too slow to be practical as the length L goes large.

For examples, suppose that a player $\mathcal{P}_X^i$ wants to optimize its action sequence $x^i \in (X^i)_L$ with length $L = 5$ over an action set $X^i = \{0, 1\}$. Assume the player's optimal action sequence $x^{i,*} \in (X^i)_5$ is all zero actions, i.e. repeating $x^{i,*} = (0, 0, 0, 0, 0)$, and the current action sequence is $x^i = (1, 0, 0, 0, 0)$. At the start of every new action sequence period, with small probability ϵ, the player experiments a uniformly random action sequence. Otherwise the player repeat his current action sequence with probability $1 - \epsilon$. Since the number of action sequence candidates is 2^5, a probability that the player chooses $x^{i,*}$ by luck is $\frac{\epsilon}{2^5}$. After choosing a trial sequence, five time steps are required to test the result of the new action sequence. Even though only one element is mismatched between $x^{i,*}$ and x^i, it can take a very long time to correct it.

To alleviate this problem, this research introduces a notion of *micro-players* which are virtual entities in actual players.Now go back to our team-based zero-sum game setting and

suppose that both $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ have the same number of micro-players. Every time step, one micro-player in each player is chosen to play an action, observes the result and optimizes its action independently. Suppose that $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ choose micro-players with the same order. Then each micro-player in $\mathcal{P}_X^1$ always plays a game with a specific micro-player in $\mathcal{P}_X^2$, and vice versa. The notion of micro-player allows a player to replace its decision in high dimensional space with multiple decisions in lower dimensions. Moreover the exact matching between micro-players in different team players allows them to correlate their actions.

Consider the previous example with the optimal sequence $x^{i,*} = (0, 0, 0, 0, 0)$ and current sequence $x^i = (1, 0, 0, 0, 0)$ again. Now assume that there are five micro-players in the player $\mathcal{P}_X^i$ and each micro-player is responsible for one element in the player's action sequence. Then the action sequence x^i can be updated to $x^{*,i}$ by adapting only one micro-player's action. In this idea, a proper utility function design for micro-players is required to selectively update the micro-player who is responsible for a bad performance of the entire sequence.

The formal setting for micro-player approach is as follows. Consider a team-based zero-sum game $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ where the opponent $\mathcal{P}_Y$'s strategy is a static k-bounded recall $\sigma \in \Sigma_k$. Assume that each of the team players has m micro-players. Let $\mathcal{I}$ be a set of ID numbers for micro players, i.e. $\mathcal{I} = \{1, 2, ..., m\}$, and $\mathcal{P}_X^{i,j}$ be j-th micro-player of $\mathcal{P}_X^i$ where $j \in \mathcal{I}$. Define $\xi^i : \mathcal{I} \to X_i$ as a function that maps a micro player $\mathcal{P}_X^{i,j}$ to an action. For a given ID sequence $s = (s_1, s_2, ..., s_L) \in (\mathcal{I})_L$, let $\xi^1(s)$ (and $\xi^2(s)$) be a periodic sequence in $(X_1)_L$ (and $(X_2)_L$) that is canonically induced from s by ξ^1 (and ξ^2).

$$\xi^1(s) = (\xi^1(s_1), \xi^1(s_2), ..., \xi^1(s_L)) \in (X_1)_L$$

$$\xi^2(s) = (\xi^2(s_1), \xi^2(s_2), ..., \xi^2(s_L)) \in (X_2)_L$$

The notation $\xi(s_j)$ and $\xi(s)$ will be used for a joint action $(\xi^1(s_j), \xi^2(s_j))$ and a joint sequence $(\xi^1(s), \xi^2(s))$ respectively. For convenience, we will allow any integer value for an

index j when denoting a j^{th} element s_j in a periodic sequence $s \in (S)_{\mathcal{I}}$. Because $s \in (S)_L$ is a periodic sequence, s_j is equivalent to s_{j+L} for every integer $j \in \mathbb{Z}$. In the rest of this chapter, we assume that the value of L, m and an ID sequence $s \in (\mathcal{I})_L$ is commonly known by the team players $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$. Therefore coordinating the team players' action sequences is equivalent to coordinating ξ^1 and ξ^2.

The following two sections introduce payoff-based learnings for periodic sequence optimization using micro-players. In the first section, the number of micro-players is the same as the period of action sequences, i.e., $m = L$. Hence each micro-player has one chance to play an action during one period. However the number of micro-player is increased as the action sequence's period goes longer. The second section reduces the number of micro-players so that the number of micro-player is much smaller than the sequence period, i.e., $m < L$. Each micro-player is allowed to play its action multiple times during one period in this setup. It is shown that one can arbitrarily increase the period of action sequences without adding more micro-players.

4.6 Micro-player matching

Suppose that m is equivalent to L. Let a ID sequence $s \in (\mathcal{I})_L$ be a periodic sequence such that

$$s = (1, 2, ..., m-1, m) \in (\mathcal{I})_L = (\mathcal{I})_m.$$

Define the utility function $U_{i,j} : (X_i)_L \times (X_{-i})_L \times \Sigma_k \to \mathcal{R}$ for a micro-player $\mathcal{P}_X^{i,j}$ as

$$U_{i,j}(\xi^i(s), \xi^{-i}(s); \sigma) = \frac{1}{k+1} \sum_{h=j}^{j+k} \mathbb{E}\phi(\xi^i(h), \xi^{-i}(h), \sigma(\xi(h-k), ..., \xi(h-1))). \quad (8)$$

Hence the micro-player $\mathcal{P}_X^{i,j}$'s utility $U_{i,j}$ is an average of rewards for the next $(1+k)$ steps. For a given opponent's strategy $\sigma \in \Sigma_k$, define $\phi_\sigma^L : (X_i)_L \times (X_{-i})_L \to \mathbb{R}$ as follows.

$$\phi_\sigma^L(\xi^i(s), \xi^{-i}(s)) = \sum_{j \in \mathcal{I}} \mathbb{E}\phi(\xi^i(j), \xi^{-i}(j), \sigma(\xi(j-k), ..., \xi(j-1))) \quad (9)$$

Note that maximizing average reward for the team $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$ is equivalent to maximizing ϕ_σ^L.

Lemma 4.6.1. *Let a utility for each micro-player is defined as (8). For a given opponent's strategy $\sigma \in \Sigma_k$, the team-based zero-sum game $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ can be transformed into a potential game of micro-players, $\hat{\mathbf{G}} = (\{\mathcal{P}_X^{i,j}\}, \{X_i\}, \{U_{i,j}\})$ for $i \in \{1, 2\}$ and $j \in \mathcal{I}$, and ϕ_σ^L is a potential function of $\hat{\mathbf{G}}$.*

Proof. Define $\xi^{i,j}(s, a)$ as an action sequence such that the j-th element of $\xi^i(s)$ is replaced with a, i.e.,

$$\begin{aligned}\xi^{i,j}(s, a) &= (\xi^i(s_1), ..., \xi^i(s_{j-1}), a, \xi^i(s_{j+1}), ..., \xi^i(s_m)) \\ &= (\xi^i(1), ..., \xi^i(j-1), a, \xi^i(j+1), ..., \xi^i(m)).\end{aligned}$$

In other words, $\mathcal{P}_X^i$'s action sequence $\xi^i(s)$ is changed to $\xi^{i,j}(s, a)$ if its j-th micro-player $\mathcal{P}_X^{i,j}$ changes its action from $\xi^i(j)$ to a.

It is trivial to prove that $U_{i,j}$ and ϕ_σ^L satisfy the following for all i, j, σ and $a \in X_i$

$$\begin{aligned}U_{i,j}(\xi^i(s), \xi^{-i}(s); \sigma) - U_{i,j}(\xi^{i,j}(s, a), \xi^{-i}(s); \sigma) \\ = \phi_\sigma^L(\xi^i(s), \xi^{-i}(s)) - \phi_\sigma^L(\xi^{i,j}(s, a), \xi^{-i}(s)).\end{aligned} \tag{10}$$

The equation (10) means that if $\mathcal{P}_X^{i,j}$ changes its current action $\xi^i(j)$ and all other micro-players in the team play with their current action mappings, then the change of $U_{i,j}$ is equivalent to the change of the potential function ϕ_σ^L. □

Because the game $\hat{\mathbf{G}}$ is a potential game, we can use learning algorithms developed for potential games. In this literature, there are several payoff-based learning algorithms that guarantee potential function maximizers are stochastically stable. If all micro-players play one of such algorithms then one can expect that a joint action sequence maximizing ϕ_σ^L is stochastically stable.

We will apply a variant of log-linear learning to adapt micro-players' actions. The

original log-linear learning rule[3] for the game $\hat{\mathbf{G}}$ can be written as the following form

$$p_a = \frac{\exp(\frac{1}{\tau}U_{i,j}(\xi^{i,j}(s,a),\xi^{-i}(s),\sigma))}{\sum_{\bar{a}_i \in \mathcal{A}_i} \exp(\frac{1}{\tau}U_{i,j}(\xi^{i,j}(s,\bar{a}),\xi^{-i}(s),\sigma))} \tag{11}$$

where p_a is a probability to play an action $a \in A_i$. However $U_{i,j}(\xi^{i,j}(s,a),\xi^{-i}(s),\sigma)$ cannot be directly calculated in our problem settings because it requires knowledge about the payoff structure and other micro-players' actions.

Therefore micro-players try to estimate $U_{i,j}(\xi^{i,j}(s,a),\xi^{-i}(s),\sigma)$ instead of calculating an exact value of it. Define $\hat{r}_t^{i,j}(a)$ as an estimated value of $U_{i,j}(\xi_t^{i,j}(s,a),\xi_t^{-i}(s),\sigma)$. The following variant of log-linear learning rule will be considered.

$$p_a = \frac{\exp(\frac{1}{\tau}\hat{r}_t^{i,j}(a))}{\sum_{\bar{a}_i \in \mathcal{A}_i} \exp(\frac{1}{\tau}\hat{r}_t^{i,j}(\bar{a}_i))} \tag{12}$$

4.6.1 Algorithm description

This section illustrates the detail of the learning algorithm with a simple ID sequence, and proves that the resulting system can achieve performance that is close to one of the optimal correlated joint strategies. The learning algorithm for a team player $\mathcal{P}_X^i$ is separated into two parts; an algorithm for an actual player $\mathcal{P}_X^i$, and another algorithm for a micro-player $\mathcal{P}_X^{i,j}$. For convenience, the algorithms are described by pseudocodes based on object-oriented design. The algorithmic flow of a repeated team-based zero-sum game $\mathbf{G}$ is illustrated in Algorithm 1. Algorithm 2 and Algorithm 3 are the algorithms for a player $\mathcal{P}_X^i$ and its micro-player $\mathcal{P}_X^{i,j}$ respectively. Note that the pseudocodes in this section are written to describe a sketchy flow of the algorithms, and these are not exhaustive descriptions. Some trivial initialization arguments or minor parameters might be omitted here.

First, a repeated team-based zero-sum game $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ are illustrated in Algorithm 1. We will assume that the opponent's rationality bound k and the period of action sequence L, which is equivalent to the number of micro-players m, are known to all team players. Therefore every team player has the same periodic ID sequence

[3] see Chapter 2

Algorithm 1 A repeated team-based zero-sum game **G**

```
teamPlayer1 = new Player(rationalityBound,sequencePeriod)   ▷ declare P_X^1
teamPlayer2 = new Player(rationalityBound,sequencePeriod)   ▷ declare P_X^2
opponent = new Opponent(rationalityBound)                   ▷ declare P_Y
t = 0                                                       ▷ time step t
while true do
    t ← t + 1
    x^1 ← teamPlayer1.playAction()          ▷ each player chooses its action to play
    x^2 ← teamPlayer2.playAction()
    y ← opponent.playAction()
    r ← φ(x^1, x^2, y)                                      ▷ r is a realized reward
    teamPlayer1.observeResult(r)            ▷ P_X^i cannot observe other players
    teamPlayer2.observeResult(r)
    opponent.observeResult(x^1, x^2, r)     ▷ P_Y can observe the team players' actions
end while
```

$s = (1, 2, ..., m)$. By Corollary 4.4.1, the value of m should be large enough to achieve a performance that is arbitrarily close to the optimal correlated strategy. Every time step, all players, $\mathcal{P}_X^1, \mathcal{P}_X^2$ and $\mathcal{P}_Y$, in the game choose their actions x^1, x^2 and y, then a reward r is determined as $\phi(x^1, x^2, y)$. The team players $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ cannot observe other players' actions and any form of communication is not allowed. Hence the input argument for *observeResult()* function of team players is only r. On the other hand, the opponent player $\mathcal{P}_Y$ can observe the team players' actions, and has three inputs x^1, x^2, r for *observeResult()* function.

Next, let's see the algorithm for a player $\mathcal{P}_X^i$ described in Algorithm 2. Each team player $\mathcal{P}_X^i$ has its own set of micro-players $\{\mathcal{P}_X^{i,1}, ..., \mathcal{P}_X^{i,m}\}$ which is denoted as *MicroPlayer[m]* in Algorithm 2. Whenever the player's *playAction()* function is called, a micro-player who will select an action for time t is determined by $s(t)$, the t^{th} element of the cyclic ID sequence s. The corresponding micro-player *MicroPlayer(s(t))* returns an action a as a result of *nextAction()*. When the player receives the realized result r of the game through *observeResult(r)*, the reward r is passed on to the last k micro-players. Therefore, from the individual micro-player's point of view, each micro-player can receive the next $(1 + k)$

Algorithm 2 algorithm for a player $\mathcal{P}_X^i$

CLASS: Player
Attributes
t: current time step
A_i: an action set
k: the opponent's rationality bound
m: a number of micro-players
s: a cyclic ID sequence
MicroPlayer[m]: a set of m MicroPlayer objects
Methods
Player(rationalityBound,sequencePeriod)
$t \leftarrow 0$
$k \leftarrow$ rationalityBound
$m \leftarrow$ sequencePeriod
$s \leftarrow (1, 2, ..., m-1, m)$
for $j = 1$ to m
MicroPlayer(j).initialization(k)
end for
a **=Player.playAction()**
$t \leftarrow t + 1$
$a \leftarrow$ MicroPlayer($s(t)$).nextAction()
return a
Player.observeResult(r)
for $j = t - k + 1$ to t
MicroPlayer($s(j)$).updateReward(r)
end for

rewards after playing an action. Note that each micro-player requires $(1 + k)$ consecutive rewards to calculate its utility function $U_{i,j}$ in (8).

Lastly, Algorithm 3 demonstrates how micro-players can learn and adapt its action. Before starting learning, a micro-player $\mathcal{P}_X^{i,j}$ initializes $\hat{r}$, which is an estimated rewards for playing each action $a \in A_i$. When a specific micro-player is selected to play an action, the selected micro-player repeats its current action a_c with probability $(1-\omega)$ or experiments a new action with probability ω. If it decides to experiment a new action, a trial action $a \in A_i$ is selected with the strategy (12). The selected action is returned to a player $\mathcal{P}_X^i$. After $\mathcal{P}_X^i$ plays an action, the micro player receives realized rewards through *updateReward()* function during the next $(k + 1)$ time steps. In *updateReward()*, two parameters c and h

Algorithm 3 algorithm for a micro-player $\mathcal{P}_X^{i,j}$

CLASS: MicroPlayer

Attributes

A_i: an action set
k: the opponent's rationality bound
ω: an experiment rate
τ: a temperature parameter for learning
$\hat{r}[|A_i|]$: a vector of estimated rewards for playing each action $a \in A_i$
h: sum of the last k reward history
c: reward counter
α: discount factor for reward estimation
$a_c \in A_i$: a current action

Methods

MicroPlayer.initialization(rationalityBound)

$k \leftarrow$ rationalityBound
for each $a \in A_i$
$\hat{r}(a) \leftarrow 0$
end for

a_c **= MicroPlayer.nextAction()**

if decide to experiment with probability ω
select a new action a_c with the following strategy p_a

$$p_a = \frac{\exp(\frac{1}{\tau}\hat{r}(a))}{\sum_{\bar{a}_i \in \mathcal{A}_i} \exp(\frac{1}{\tau}\hat{r}(\bar{a}_i))}$$

else
$a_c \leftarrow a_c$ ▷ repeat a current action a_c
end if
$c \leftarrow 0$
$h \leftarrow 0$
return a_c

MicroPlayer.updateReward(r)

$c \leftarrow c + 1$
$h \leftarrow h + r$
if $c = k + 1$ ▷ if $(k+1)$ rewards are received after the current action
$\hat{r}(a_c) \leftarrow (1 - \alpha)\frac{h}{k+1} + \alpha \cdot \hat{r}(a_c)$
end if

are updated to calculate its realized utility $U_{i,j}$ in (8). The parameter c counts a number of rewards after playing an action and h is a sum of those rewards. When $(k+1)$ rewards are observed ,i.e., $c = k+1$, then the utility estimation $\hat{r}^{i,j}(a_c)$ for its current action a_c is updated with discount factor α.

$$\hat{r}^{i,j} + t + 1(a_c) \leftarrow (1-\alpha)\frac{h}{k+1} + \alpha \cdot \hat{r}_t^{i,j}(a_c)$$

Then one can easily prove the following lemma regarding estimation error.

Lemma 4.6.2. *Let $\hat{r}_t^{i,j}(a)$ be a micro-player $\mathcal{P}^{i,j}$'s estimated utility for playing a at time t. If the micro-players' experiment rate ω is small enough, then the error of utility estimation is bounded with some value δ_2 for sufficiently large time t,i.e.,*

$$|\hat{r}_t^{i,j}(a) - U_{i,j}(\xi_t^{i,j}(s,a)\xi_t^{-i}(s), \sigma)| < \delta_2$$

The following theorem proves that micro-player matching described above can achieve correlated behavior.

Theorem 4.6.1. *Let $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ be a team-based zero-sum game. Let the opponent player $\mathcal{P}_Y$ have a static k-recall strategy $\sigma \in \Sigma_k$. Let the team players, $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$, coordinate their m-periodic action sequences with the micro-player matching algorithm illustrated in this section. Suppose that the experiment rate ω is small enough. Then, for sufficiently large time T and sequence length m, the average reward for the team players' periodic action sequences are arbitrarily close to performance of the optimal correlated joint strategy, i.e.,*

$$\max_{p \in \Delta(X_1 \times X_2)} \phi(p, \sigma) - \epsilon < \frac{1}{m} \sum_{t=T+1}^{T+m} \phi(x^1(t), x^2(t), y(t))$$

where $\epsilon > 0$ is arbitrarily small

Proof. Assume that m is large enough so that Corollary 4.4.1 guarantees existence of a joint action sequence $x = (x^1, x^2) \in (X^1 \times X^2)_m$ such that, for some $\epsilon_1 > 0$,

$$\max_{p \in \Delta(X^1 \times X^2)} \phi(p, \sigma) - \epsilon_1 < \phi_\sigma^m(x^1, x^2).$$

Define $(\xi^{1,*}(s), \xi^{2,*}(s))$ as a joint action sequence of the team players that maximizing ϕ_σ^m, i.e.,

$$(\xi^{1,*}(s), \xi^{2,*}(s)) = \underset{(x^1,x^2)\in(A_1\times A_2)_m}{\arg\max} \phi_\sigma^m(x^1, x^2).$$

Then the following is true.

$$\max_{p\in\Delta(X^1\times X^2)} \phi(p,\sigma) - \epsilon_1 < \phi_\sigma^m(\xi^{1,*}(s), \xi^{2,*}(s))$$

Lemma 4.6.1 proved that the game $\mathbf{G}$ can be transformed into a potential game of micro-players $\hat{\mathbf{G}} = (\{\mathcal{P}_X^{i,j}\}, \{X_i\}, \{U_{i,j}\})$ and ϕ_σ^m is a potential function of $\hat{\mathbf{G}}$. If all micro-players could play the original form of log-linear learning like (11), then the joint action sequence $(\xi^{1,*}(s), \xi^{2,*}(s))$ is stochastically stable by [12].

However, as described in (12), micro-player matching employs estimated rewards $\hat{r}_t^{i,j}$ instead of the utility $U_{i,j}$. Let P^{ϵ_2} be the perturbed Markov process induced from the original log-linear learning in (11) which is based on the exact value of utility. Similarly, let $P(t)$ be the perturbed Markov process induced from micro-player matching at time t. By Lemma 4.6.2, a distance between $P(t)$ and P^{ϵ_2} can be bounded for sufficiently large time t, i.e.,

$$\limsup_{t=0,1,2\ldots} \|P(t) - P^{\epsilon_2}\| < \delta_3.$$

Then the rest of proof is easily induced from Corollary 3.2.1. □

4.6.2 Simulations

This section presents two examples to test micro-player matching and the simulation results are shown.

4.6.2.1 Example 1

In this simulation, the action sets were $X_1 = X_2 = Y = \{0, 1\}$ and the opponent player $\mathcal{P}_Y$'s bounded recall strategy order had an order $k = 2$. The opponent's action selection rule did not depend on $\mathcal{P}_X^2$'s action, and it was only conditioned on $\mathcal{P}_X^1$'s action history during previous $k = 2$ time steps as shown in Table 1. The action sequence period of the

Table 1: Action selection rule for the opponent ($\mathcal{P}_Y$)

$x^1(t-2)$	$x^1(t-1)$	$y(t)$
0	0	1
0	1	1
1	0	0
1	1	0

Table 2: An example of the team's optimal sequences.

$x^1(t)$	...	0	0	1	1	0	0	1	1	...
$x^2(t)$	...	0	0	1	1	0	0	1	1	...
$y(t)$	...	0	0	1	1	0	0	1	1	...
$\phi(x^1(t), x^2(t), y(t))$	...	1	1	1	1	1	1	1	1	...

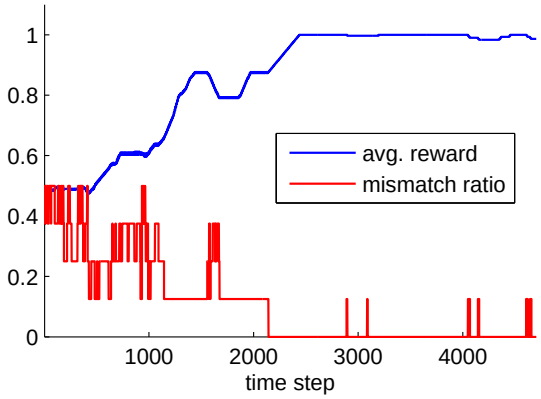
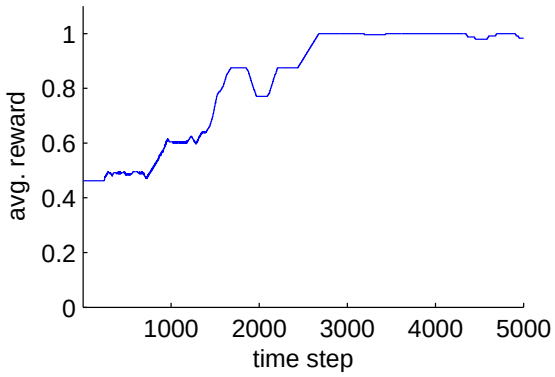


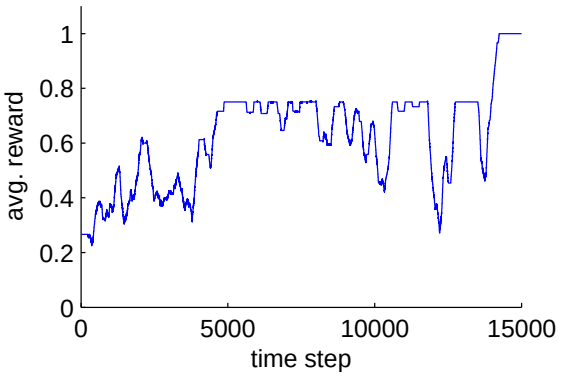
Figure 12: Simulation result of sequence coordination example 1

team $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$ was $L = m = 8$. Then one of the team's optimal action sequences that maximizing average $\phi(x^1(t), x^2(t), y(t))$ is 00110011 as Table 2, and the maximum average reward for team players is 1. In the simulation, the probability to update an action sequence was $\omega = 0.2$ for both $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$. The exploration parameter $\frac{1}{\tau}$ for action selection (12) was 4 .

Figure 12 illustrates a simulation result. The blue line is an average reward for the team during last 300 time steps, and the red line is a ratio of mismatched elements between sequences of $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$. For an example, if the teams' action sequences of length 8, x_1 and



(a) micro-player matching



(b) Payoff-based log-linear learning without introducing micro-players

Figure 13: Simulation results for learning speed comparison. The micro-player matching could coordinate about five times faster than the case in which the team players played payoff-based log-linear learning without introducing a notion of micro-players.

x_2, are (00001111) and (00001100) respectively, then the mismatch ratio is $\frac{2}{8}$. Therefore if every micro-player in the team players chooses the same action as its partner micro-player, the mismatch ratio becomes 1.

The simulation result shows that ,after about 2100 time steps, $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ successfully coordinated their action sequences and their average reward reached the maximum reward 1. Even after the successful coordination, micro-players within the team players sometimes experimented random actions so that one can observe small noise in the mismatch ratio graph.

Another simulation was conducted to compare learning speeds of the micro-player matching and a general payoff-based learning in which an entire action sequence is treated as one object. The payoff-based log-linear learning from [12] applied in this simulation is a variant of original log-linear learning and its mechanism is very similar to the proposed micro-player matching except the fact that a L-length action sequence is dealt as one action. In the payoff-based log-linear learning, at every end of an action sequence period, each team player $\mathcal{P}_X^i$'s action sequence is selected among the set of L-length action sequences, $(X^i)_L$. The game settings including the opponent's strategy for the simulation

was the same as the previous simulation. All players have an action set $\{0, 1\}$ and action sequence period was $L = 8$.

Figure 13 (a) and (b) are the average rewards over time in the micro-player matching and the payoff-based log-linear learning respectively. Note that the ranges of X-axis for the two simulation results in Figure 13 are different. While the micro-player matching could converge to the optimal average reward after less than 3000 time steps, the payoff-based log-linear learning took about 15,000 time steps until converging to the optimal average reward. Therefore in this simulation, the team players could save about 80% of time for coordinating their action sequences by introducing the micro-player concept.

4.6.2.2 Example 2

Let's see another example. Suppose that action sets were $X_1 = X_2 = Y = \{0, 1\}$, and the opponent player $\mathcal{P}_Y$'s strategy order was $k = 3$. The length of action sequences for $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ was $L = m = 16$.

The opponent $\mathcal{P}_Y$ makes a decision using a fictitious play [4] with k time step memory. Fictitious play is a class of the well-known learning algorithm based on empirical frequency of observed actions during the last k-time steps. The opponent selects its action as follows.

$$y(t) = \begin{cases} 0 & \text{if } \frac{1}{2k} \sum\limits_{h \in \{t-k,\ldots,t-1\}} (x^1(h) + x^2(h)) > 0.5 \\ 1 & \text{if } \frac{1}{2k} \sum\limits_{h \in \{t-k,\ldots,t-1\}} (x^1(h) + x^2(h)) < 0.5 \end{cases}$$

If each 0 and 1 occurs exactly k times, i.e., $\frac{1}{2k} \sum\limits_{h \in \{t-k,\ldots,t-1\}} (x^1(h) + x^2(h)) = 0.5$, then the opponent selects its action with uniformly random distribution. It can be interpreted as that if one of actions among $\{0, 1\}$ is more frequently selected by $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$, then the opponent $\mathcal{P}_Y$ chooses the opposite action. Note that this $\mathcal{P}_Y$'s model with fictitious play is a mixed k-recall strategy $\sigma \in \Sigma_k$.

This is a good example to compare the performance of coordinated joint action sequences with independently mixed strategies. Similar to the maxmin payoff comparison in

[4] see Chapter 2 for more details about a fictitious play.

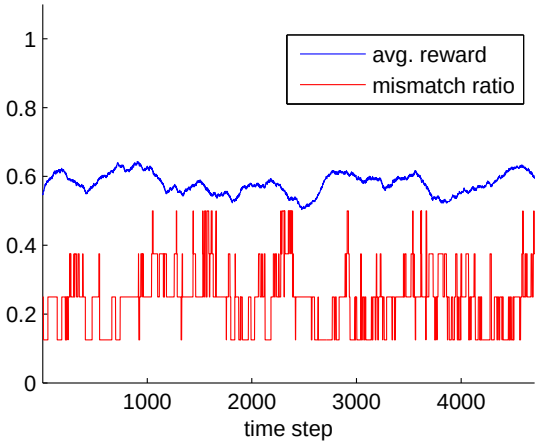


Figure 14: Simulation result of sequence coordination example 2

Section 4.3, if each $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ independently selects its one-shot strategy without communications, the optimal joint strategy μ_{ind}^* will be

$$\mu_{\text{ind}}^* = \begin{pmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

and its expected payoff is $\frac{1}{4}$. On the other hand, the expected payoff for the optimal correlated strategy is $\frac{1}{2}$ where the optimal joint strategy μ_{cor}^* is.

$$\mu_{\text{cor}}^* = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$

The simulation result is shown in Figure 14. The probability to update an action sequence was $\omega = 0.2$ for both $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$. The discount factor α and exploration parameter $\frac{1}{\tau}$ were chosen as 0.5 and 4 respectively. The blue line is average rewards for the team $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$during last 300 time steps, and the red line is ratio of mismatched elements between the action sequences of $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$.

In the simulation results, the average reward is 0.58 which is close to the performance of correlated strategies. Therefore the resulting joint action sequence of the micro-player matching showed correlated behavior as the equivalent correlated joint strategies.

4.7 Micro-player matching with a de Bruijn ID sequence

The previous section showed that the team players can coordinate their action sequences using the concept of micro-players. However a number of micro-players, m, was the same as a length of the action sequences, L, and one can easily expect that learning speed would be impractically slow for a large number of micro-players. This section discusses how to reduce the number of micro-players for long action sequences.

The micro-player matching described in the previous section assumed that the order of micro-players is defined by a serial ID sequence $s = (1, 2, ..., m-1, m) \in (\mathcal{I})_m$. Hence each micro-player plays an action once a period. Now assume that micro-players are allowed to play an action more than once during a period. Then a number of micro-players, can be less than a length of action sequence, i.e., $m < L$. However the order of micro-players $s \in (\mathcal{I})_L$ cannot be any arbitrary sequence in this case. If s is not 'complex' enough, then its pattern will be recognized by opponents with bounded rationality.

For example, consider the illustrative team-based zero-sum game $\mathbf{G}$ from Section 4.3 again. Suppose that the opponent has 1-bounded recall strategy $\sigma \in \Sigma_1$. Both the team players $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ have two micro players, i.e., $m = 2$ so that the ID set of micro-player is $\mathcal{I} = \{1, 2\}$. Select a period of team players' action sequences as $L = 8$. Assume that the team players $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ coordinate a joint action sequence based on a given ID sequence $s \in (\mathcal{I})_8$ such that

$$s = (1, 1, 1, 1, 2, 2, 2, 2).$$

Then the team's maxmin average payoff ϕ_σ^L for a periodic action sequence $(\xi^1(s), \xi^2(s))$ is

$$\max_{\xi^1, \xi^2} \min_{\sigma \in \Sigma_1} \phi_\sigma^L(\xi^1(s), \xi^2(s)) = \frac{1}{4}$$

which is equivalent to the maxmin payoff for independently mixed joint strategy μ_{ind}^* in Section 4.3. The original purpose of introducing a micro-player is building an action sequence whose average payoff is better than non-correlated strategy μ_{ind}^*. But in this example, the pattern of the sequence s is too simple to trick the opponent's strategy σ and the average

payoff is not improved from μ^*_{ind}.

To solve the complexity issue, an ID sequence s will be selected from de Bruijn sequences [5] which is a special class of periodic sequences. Consider $s \in \mathcal{B}(m, k+1) \subset (\mathcal{I})_{m^{k+1}}$, which is a de Bruijn sequence of a symbol set $\mathcal{I}$ with order $(k+1)$. Then the period length of an action sequence x^i for $\mathcal{P}^i_X$ is m^{k+1}, i.e.,

$$x^i = \xi^i(s) = (\xi^i(s_1), \xi^i(s_2), ..., \xi^i(s_{m^{k+1}})) \in (X^i)_{m^{k+1}}.$$

Suppose that $\mathcal{P}^1_X$ and $\mathcal{P}^2_X$ share the same de Bruijn sequence $s \in \mathbb{B}(m, k+1)$. Then the joint action sequence $(\xi^1(s), \xi^2(s))$ satisfies the following characteristic for randomly selected index $\hat{t}$,

$$\mathbf{Pr}\left[\xi(s_{\hat{t}}) | \xi(s_{\hat{t}-1}), ..., \xi(s_{\hat{t}-k})\right] = \mathbf{Pr}\left[\xi(s_{\hat{t}})\right]. \tag{13}$$

In other words, the last k-steps history does not give any information about the probability of the team players' next joint action $\mathbf{Pr}\left[\xi(s_t)\right]$, and such de Bruijn sequences cannot be recognized by k-bounded recall strategies. Therefore it looks a jointly random action from the opponent's point of view.

Now, we will redefine micro-players' utility function $U_{i,j}$ for this de Bruijn ID sequence approach. Let $\xi^i(\mathcal{I})$ be a joint action profile of micro-players within a player $\mathcal{P}^i$. For every de Bruijn ID sequence $s \in \mathbb{B}(m, k+1) \in (\mathcal{I})_{m^{k+1}}$, a utility function $U_{i,j} : \prod_{i=1}^m (X^1 \times X^2) \times \Sigma_k \to \mathcal{R}$ for a micro-player $\mathcal{P}^{i,j}$ is

$$U_{i,j}(\xi^i(\mathcal{I}), \xi^{-i}(\mathcal{I}); \sigma) = \frac{1}{m^k} \sum_{\lambda \in \prod_{h=1}^k \mathcal{I}} \phi(\xi^i(j), \xi^{-i}(j), \sigma(\xi(\lambda))) \tag{14}$$

where $\xi(\lambda)$ is a k-length joint action sequence induced from a k-length ID word $\lambda \in \prod_{h=1}^k \mathcal{I}$. Note that the utility function (14) can be interpreted as an average reward for j^{th} micro-player's play. As long as the ID sequence s is a de Bruijn sequence, the utility function is independent from a selection of s because of the characteristic of de Bruijn sequences; all available k-length history of micro-player ID happens exactly once during a period from each micro-player's point of view.

[5] See Definition 4.2.2

Then it can be easily verified that, for a given static opponent's strategy $\sigma \in \Sigma_k$, the team's average reward ϕ_σ^L in (9) is a sum of each micro-players utility $U_{i,j}$. For $\forall i \in \{1,2\}$ and every de Bruijn sequence $s \in \mathcal{B}(m, k+1)$,

$$\phi_\sigma^L(\xi^i(s), \xi^{-i}(s)) = \sum_{j \in \mathcal{I}} U_{i,j}(\xi^i(\mathcal{I}), \xi^{-i}(\mathcal{I}); \sigma). \tag{15}$$

where the sequence length L is m^{k+1}.

However Theorem 4.6.1 no longer holds for this de Bruijn ID sequence approach. In micro-player matching with a serial ID sequence in the previous section, the utility function $U_{i,j}$ defined as (8) satisfies the equation (10). Accordingly the resulting game could be converted to a potential game as proved in Lemma 4.6.1. On the other hand, the redefined utility (14) cannot satisfy the equation (10), and thus learning algorithms for potential games cannot be applied in this section. The learning algorithm for a de Bruijn ID sequence approach will be illustrated after discussing the phase difference issue of this method.

Lemma 4.7.1. *Consider a team-based zero-sum game* $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$. *Suppose that the team players play the micro-player matching with de Bruijn ID sequence. Let a utility for each micro-player is defined as (14). For a given opponent's strategy* $\sigma \in \Sigma_k$*, the team-based zero-sum game* $\mathbf{G}$ *can be transformed into a normal form game of micro-players,* $\hat{\mathbf{G}} = (\{\mathcal{P}_X^{i,j}\}, \{X_i\}, \{U_{i,j}\})$ *for* $i \in \{1,2\}$ *and* $j \in \mathcal{I}$.

A significant advantage of the de Bruijn ID sequence approach is that the size of the game is independent from the rationality level of opponents. The size of the game in micro-player matching method is increased as the number of micro-players goes larger. In the previous section, the larger opponent's rationality level k requires the longer length of simple ID sequence $s = (1, 2, ..., m-1, m)$ which is equivalent to the number of micro-players . On the other hand, a de Bruijn ID sequence $s \in \mathcal{B}(m, k+1)$ can be constructed for any arbitrary positive integer k, and the value of k is independent from m. Consequently, the team players does not have to increase their number of micro-players to coordinate their behavior against *smarter* opponents with a larger rationality level.

4.7.1 Phase difference between action sequences

Even though complexity issue could be alleviated with de Bruijn sequences, the micro-player matching with a reduced number of micro-players has another problem; phase difference between team players. A phase difference between team players can damage the team's payoff for two reasons.

First, joint action sequences with phase difference can be predicted by opponents even with bounded rationality. Note that opponents with limited rationality can take advantage from its observation if the team players' action sequences do not satisfy (13). With the same de Bruijn ID sequence $s \in \mathcal{B}(m, k+1)$, if the team players' action sequences $x^1 \in (X^1)_{m^{k+1}}$ and $x^2 \in (X^2)_{m^{k+1}}$ are repeated without any phase difference, the joint sequence $(x^1(t), x^2(t))$ cannot be recognized by the opponent. However if there exists phase difference, the resulting joint action sequence does not hold (13) and k-step history provides information about the team's next action.

Secondly, phase difference hinders correlation between micro-players. Without phase difference, a micro-player $\mathcal{P}_X^{i,j}$ can correlates its action with its 'game partner' $\mathcal{P}_X^{-i,j}$ because each micro-player has a fixed partner within another team player, i.e., i^{th} micro-player in $\mathcal{P}_X^1$ always plays a game with i^{th} micro-player in $\mathcal{P}_X^2$. However $\mathcal{P}_X^{i,j}$ does not have a fixed partner if there exists a phase difference between players' action sequences. For examples, consider the de Bruijn ID sequence of an ID set $\mathcal{I} = \{1, 2\}$

$$s = (1, 1, 1, 2, 1, 2, 2, 2) \in \mathcal{B}(2, 3).$$

Assume that $\mathcal{P}_X^1$ plays one-step lagged action relative to $\mathcal{P}_X^2$, then the resulted joint sequence is as follows.

$\mathcal{P}_X^1$'s action	...	$\xi^1(1)$	$\xi^1(1)$	$\xi^1(1)$	$\xi^1(2)$	$\xi^1(1)$	$\xi^1(2)$	$\xi^1(2)$	$\xi^1(2)$	...
$\mathcal{P}_X^2$'s action	...	$\xi^2(1)$	$\xi^2(1)$	$\xi^2(2)$	$\xi^2(1)$	$\xi^2(2)$	$\xi^2(2)$	$\xi^2(2)$	$\xi^2(1)$	...

In this example, $\mathcal{P}_X^{i,j}$ encounters both $\mathcal{P}_X^{-i,1}$ and $\mathcal{P}_X^{-i,2}$ with the same empirical frequency. If a micro-player $\mathcal{P}_X^{i,j}$'s partner is not fixed to a specific micro-player $\mathcal{P}_X^{-i,j}$, then periodic

action sequence approach loses its merit relative to independently mixed action strategies. Therefore a proper phase-lock process is needed for the micro-player matching with a de Bruijn ID sequence.

Before discussing a phase-lock process, we will define a notation regarding a circular shift of a periodic sequence. Define $s^{\mathrm{sh}(\alpha)}$ as a circular shift of a periodic sequence $s \in (\mathcal{I})_L$. For an example, consider a periodic sequence $s = (1, 2, 3, 4)$ then $s^{\mathrm{sh}(\alpha)}$ is as follows.

$s^{\mathrm{sh}(2)}$	...	3	4	1	2	...
$s^{\mathrm{sh}(1)}$	...	4	1	2	3	...
$s^{\mathrm{sh}(0)} = s$	...	**1**	**2**	**3**	**4**	...
$s^{\mathrm{sh}(-1)}$	...	2	3	4	1	...
$s^{\mathrm{sh}(-2)}$	...	3	4	1	2	...

In other words, a circular shift $s^{\mathrm{sh}(\alpha)}$ is a lagging sequence if α is a positive number, and a leading sequence if α is a negative number.

Consider a periodic sequences s' which is a circular shift of a periodic sequence $s \in (\mathcal{I})_L$. Since s is a cyclic sequence, one can express s' as either a lagging sequence or a leading sequence comparing to s. If s' is expressed as an α-step lagging sequence, $s^{\mathrm{sh}(\alpha)}$, then it can be also expressed as a $(\alpha - L)$-step leading sequence, $s^{\mathrm{sh}(\alpha-L)}$. For convenience, we will denote the circular shift sequence s' as $s^{\mathrm{sh}(j)}$ if $|j| < |j - L|$ or $s^{\mathrm{sh}(j-L)}$ otherwise.

Now assume that s is a de Bruijn sequence and consider the micro-player matching with de Bruijn sequence s again. Without loss of generality, if there is phase difference, let the lagged player among the two team players is $\mathcal{P}_X^1$ and the leading player is $\mathcal{P}_X^2$. Therefore, for a given micro-player ID sequence s, the team players' action at time t can be expressed as follows.

$$\begin{aligned} x^1(t) &= \xi^1(s(t)) \\ x^2(t) &= \xi^2(s^{\mathrm{sh}(-\alpha)}(t)) \end{aligned} \tag{16}$$

And one can assume that α is always non-negative value because x^2 is a leading sequence. Define $h_k(t) \in \mathcal{I}^k$ be the k-length history of ID sequence s at time t, i.e,

$$\begin{aligned} h_k(t) &= (s(t-k), s(t-k+1), ..., s(t-1)) \\ &= (s^{\mathrm{sh}(k)}(t), s^{\mathrm{sh}(k-1)}, ..., s^{\mathrm{sh}(1)}(t)) \end{aligned} \tag{17}$$

Note that k-step history of a sequence $s(t) \in \mathcal{I}$ at time t can be expressed as a joint of lagging sequences. Then the opponent $\mathcal{P}_Y$'s k-recall strategy σ is

$$\sigma(\xi^1(h_k(t)), \xi^2(h_k(t+\alpha))) = \sigma(\xi^1(h_k(t)), \xi^2(h_k(t+\alpha))) \tag{18}$$

where the team players' k-step action histories $\xi^1(h(t))$ and $\xi^2(h(t+\alpha))$ are defined as follows.

$$\begin{aligned} \xi^1(h_k(t)) &= \xi^1(s^{\mathrm{sh}(k)}(t), s^{\mathrm{sh}(k-1)}(t), ..., s^{\mathrm{sh}(1)}(t)) \\ \xi^2(h_k(t+\alpha)) &= \xi^2(s^{\mathrm{sh}(k-\alpha)}(t), s^{\mathrm{sh}(k-1-\alpha)}(t), ..., s^{\mathrm{sh}(1-\alpha)}(t)) \end{aligned} \tag{19}$$

The main idea for phase difference detection is that $\mathcal{P}_X^1$'s realized rewards $r(t) = \phi(x^1(t), x^2(t), y(t))$ is somehow associated with some leading ID sequence $s^{\mathrm{sh}(-\alpha)}$ if and only if $\mathcal{P}_X^2$'s sequence is leading: in the equations (16), (18) and (19), a shifted sequence $s^{\mathrm{sh}(j)}$ with negative value j appears if and only if $\mathcal{P}_X^2$'s sequence is leading. On the other hand, the team players' action histories (19) are related to a joint of lagging sequences regardless the existence of phase difference. Hence a reward sequence is always associated with the lagging sequence, and the leading player cannot determine whether there exists a lagging player by observing correlation between $r(t)$ and any lagging sequence $s^{\mathrm{sh}(j)}$. But the lagging player can observe correlation between leading sequences and the reward sequence when $\alpha \neq 0$. Therefore lagging player can determine whether he is lagged behind his team player or not. If a player observes strong correlation between $s^{\mathrm{sh}(-\alpha)}(t)$ and $r(t)$, it skips α steps from its current ID sequence to align with its leading team player.

To apply the method described above, a proper metric is required to measure the correlation between a reward sequence and shifted ID sequences. Note that we cannot assume

linear property on the reward function $\phi(x^1, x^2, y)$, and the players' actions can be nominal values. Consequently most of conventional correlation coefficient for real valued data, such as Pearson's correlation coefficient, cannot be used as a measure for our method. In this research, the uncertainty coefficient [72] will be used as a metric that measures correlation between sequences. The uncertainty coefficient is based on entropy from information theory literature and can be calculated for nominal data.

Definition 4.7.1 (Theil, 1970 [72])**.** *Consider two discrete random variables A and B. The uncertainty coefficient $\mathcal{U}(B|A)$ is*

$$\mathcal{U}(B|A) = \frac{H(B) - H(B|A)}{H(B)} = \frac{I(A;B)}{H(B)}.$$

$\mathcal{U}(B|A)$ is a normalized mutual information $I(A;B)$. By the definition, a value of $\mathcal{U}(B|A)$ lies between 0 and 1. Note that the uncertainty coefficient is not symmetric, i.e., $\mathcal{U}(B|A) \neq \mathcal{U}(A|B)$. The uncertainty coefficient $\mathcal{U}(B|A)$ gives the fraction of uncertainty of B that is lost if A is already known. If $\mathcal{U} = 0$, A and B are independent. If $\mathcal{U} = 1$, B is completely predictable by A. For in-between values, one should cautiously interpret the uncertainty coefficient. For examples, suppose that the value of $\mathcal{U}(B|A)$ is 0.3 for some random variables A and B. Are these two variables A and B associated or not? There is no general threshold regarding $\mathcal{U}(B|A)$ that guarantees whether the variables are associated or not. In micro-player matching, this threshold is heuristically chosen according to the given game. The detail of phase-lock process is illustrated in the next section.

4.7.2 Algorithm description

This section illustrates algorithmic details of the micro-player matching with a de Bruijn ID sequence. There are two major concerns in the algorithm; a phase-lock process and sequence optimization. An individual player performs phase-lock process every end of its action sequence period using uncertainty coefficient. For sequence optimization, the regret testing rule [29] is applied to each micro-player.

Before discussing detail of the algorithm, we will put a few assumptions on the system. First, each $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ has its own independent time clock. This clock shows the flow of discrete time step t and both players choose their actions $\xi^i(s_t)$ based on their individual time clocks. Second, all players' time clocks start at the same time with the same initial value $t_0 = 0$, so there is no phase difference in normal cases. Lastly, every time tick, there exists very small probability ϵ that the clocks fails to increase t due to some external factors. If $\mathcal{P}_X^1$ fails to increase its time clock then his phase lagged one-step behind comparing to $\mathcal{P}_X^2$ and vice versa. With this mechanism, the team players' phase difference varies slowly and phase difference cannot be increased or decreased more than one during one time step. Therefore if the team players can align their phases properly then the phase difference would not be a large number. In our algorithm, it is assumed that the amount of phase difference α between team players is maintained less than some small value $\bar{\alpha}$.

Similar to Section 4.6.1, object-oriented pseudocodes are written to describe the micro-player matching with de Bruijn sequence. The algorithmic flow of a repeated team-based zero-sum game $\mathbf{G}$ is omitted here since it is identical to Algorithm 1. The algorithms for a player $\mathcal{P}_X^i$ and its micro-player $\mathcal{P}_X^{i,j}$ are Algorithm 4 and Algorithm 5 respectively.

The player $\mathcal{P}_X^i$'s algorithm in Algorithm 4 is similar to Algorithm 2 from the previous section except the part related to a phase-lock process. Using the uncertainty coefficient, the phase-lock process for the micro-player matching is as follows. At every end of its action sequence period, *phaseDifferenceCheck()* function is called. Then the team player $\mathcal{P}_X^i$ tests the uncertainty correlation

$$\mathcal{U}(r|s^{\text{sh}(-j)}) = \frac{I(r; s^{\text{sh}(-j)})}{H(r)} \tag{20}$$

between r, the observed reward samples during the period and $s^{\text{sh}(-j)}$, leading ID sequences for $j \in \{1, ..., \bar{\alpha}\}$. For entropy calculation, probability densities are calculated from observed samples during the last one period of action sequences (i.e. m^{k+1}-steps). If $\mathcal{U}(r|s^{\text{sh}(-j)})$ is larger than a heuristically chosen threshold $\bar{u}$ then we think that r and $s^{\text{sh}(-j)}$ are associated. In Algorithm 4, $\bar{\alpha}$ and $\bar{u}$ are selected as 1 and 0.15 respectively, thus it test

Algorithm 4 algorithm for a player $\mathcal{P}_X^i$

CLASS: Player
Attributes
t: current time step
A_i: an action set
k: the opponent's rationality bound
m: a number of micro-players
s: a De Bruijn ID sequence $\mathcal{B}(m, k+1)$
$\bar{u}$: a uncertainty coefficient threshold
MicroPlayer[m]: a set of m MicroPlayer objects

Methods
Player(rationalityBound,sequencePeriod)
$t \leftarrow 0$
$k \leftarrow$ rationalityBound
$m \leftarrow$ sequencePeriod
$s \leftarrow \mathcal{B}(m, k+1)$
for $j = 1$ to m
MicroPlayer(j).initialization(k)
end for
a **=Player.playAction()**
$t \leftarrow t + 1$
$a \leftarrow$ MicroPlayer($s(t)$).nextAction()
return a
Player.observeResult(r)
for $j = t - k + 1$ to t
MicroPlayer($s(j)$).updateReward(r)
end for
if t mod L == 0 ▷ if it is the end of current period
for $j = t - k + 1$ to t
MicroPlayer($s(j)$).updateAction() ▷ perform regret testing
end for
α=phaseDifferenceCheck() ▷ check phase difference
if $\alpha < 0$ ▷ if its ID sequence is lagging, then adjust the index t
$t \leftarrow t - \alpha$
end if
end if
α **= Player.phaseDifferenceCheck()**
$\alpha \leftarrow 0$
if $H(r) > 0.1$ and $\mathcal{U}(r; s^{\text{sh}(-1)}) > \bar{u}$
$\alpha \leftarrow -1$
end if
return α

whether $\mathcal{U}(r|s^{\text{sh}(-1)})$ is larger than 0.15 or not.

However the entropy $H(r)$ should be larger than a certain amount to test 'meaningful correlation' between s and r. For examples, consider a team of two players with the micro-player matching and assume that there is no phase difference between the two players. Suppose that all micro-players in both team players could not coordinate their actions so that the observed reward $r(t)$ are all zeros. The empirical entropy of $r(t)$ is $H(r) = 0$ and consequently, $I(r; s^{\text{sh}(j)})$ is also zero for any j. Therefore there is a condition $H(r) > 0.1$ in *phaseDifferenceCheck()*.

Algorithm 5 illustrates a micro-player $\mathcal{P}_X^{i,j}$'s mechanism which is a variant of regret testing [29, 31][6]. During initialization stage, parameters $\hat{r}$ and c which are related to average rewards calculation are reset and a base action a_b is randomly selected among $\mathcal{A}_i$. Whenever the specific micro-player $\mathcal{P}_X^{i,j}$ is selected to play an action for $\mathcal{P}_X^i$, the micro-player has two options. With a small experiment probability ω, the current action a_c is selected uniformly random, or a_c is its base action a_b with a probability $(1 - \omega)$. Then the realized reward r is passed on to the micro-player through *UpdateReward(r)*. To calculate the average of realized rewards for each action, the counter $c(a_c)$ is increased by one and the reward r is added to $\hat{r}(a_c)$.

If it is the end of sequence period, *UpdateAction()* functions of all micro-player are called, and each micro-player updates its base action as follows. First, it calculates *baseReward* which is an average reward for playing the base action a_b during the last period. Then compare it with *regret(a)* which is a difference between *baseReward* and an average reward of an action $a \in \mathcal{A}_i \setminus a_b$. If *regret(a)* is larger than a tolerance level τ then a is a candidate for a new base action selection. If there is at least one candidate action, the new base action

[6] See Chapter 2 for more detail about regret testing rules.

Algorithm 5 algorithm for a micro-player $\mathcal{P}_X^{i,j}$

CLASS: MicroPlayer

Attributes

$\mathcal{A}_i$: an action set
ω: a rate to experiment an action other than a_b
β: a temperature parameter for learning
τ: tolerance level
λ: a rate to update a_b even there is no action with large regret
$\hat{r}[|\mathcal{A}_i|]$: sum of rewards for playing $a \in \mathcal{A}_i$
$c[|\mathcal{A}_i|]$: counter for playing $a \in \mathcal{A}_i$
$a_c \in A_i$: a current action
$a_b \in A_i$: a base action

Methods

MicroPlayer.Initialization()
$\hat{r}(a) \leftarrow 0$ for all $a \in \mathcal{A}_i$
$c(a) \leftarrow 0$ for all $a \in \mathcal{A}_i$
$a_b \leftarrow \text{rand}(\mathcal{A}_i)$ ▷ select a random base action a_b

a_c **= MicroPlayer.NextAction()**
if decide to experiment with a probability ω
$a_c \leftarrow \text{rand}(\mathcal{A}_i)$ ▷ select a uniformly random action a_c
else
$a_c \leftarrow a_b$ ▷ repeat the base action a_b
end if
return a_c

MicroPlayer.UpdateReward(r)
$c(a_c) \leftarrow c(a_c) + 1$
$\hat{r}(a_c) \leftarrow \hat{r}(a_c) + r$

MicroPlayer.UpdateAction()
baseReward $\leftarrow \frac{\hat{r}(a_b)}{c(a_b)}$ ▷ average reward for the base action a_b
for each $a \in \mathcal{A}_i \backslash a_b$
regret$(a) \leftarrow \frac{\hat{r}(a)}{c(a)}$−baseReward ▷ regret for not playing a
if regret$(a) > \tau$
$p_a \leftarrow \exp(\beta \cdot \text{regret}(a))$
else
$p_a \leftarrow 0$
end if
end for
$\hat{r}(a) \leftarrow 0$ for all $a \in \mathcal{A}_i$ ▷ reset parameters for the next period
$c(a) \leftarrow 0$ for all $a \in \mathcal{A}_i$
if at least one p_a is non-zero
select a_b with a normalized probability vector p_a
else
with a probability λ, select a uniformly random a_b
otherwise, keep the current a_b
end if

is selected according to the following strategy.

$$p_a = \frac{\exp(\beta \cdot \textit{regret}(a))}{D}, \qquad \text{if } \textit{regret(a)} > \tau$$
$$p_a = 0, \qquad \text{otherwise} \tag{21}$$

where D is a normalization factor and β is a temperature parameter for learining. If *regret(a)* for every action $a \in \mathcal{A}_i \setminus a_b$ is smaller than τ then the micro-player selects a uniformly random base action a_b with a small probability λ or repeat the current base action a_b with a probability $(1 - \lambda)$.

The following theorem is the direct consequence of Lemma 4.7.1 and Theorem 2.3.3.

Theorem 4.7.1. *Let* $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ *be a team-based zero-sum game. Suppose that the opponent player* $\mathcal{P}_Y$ *have a static bounded recall strategy* $\sigma \in \Sigma_k$*. Let the team players* $\mathcal{P}_X^1$ *and* $\mathcal{P}_X^2$ *coordinate their periodic action sequences with the micro-player matching algorithm with a de Bruijn ID sequence* $s \in \mathcal{B}(m, k+1) \subset (\mathcal{I})_{m^{k+1}}$*. There are upper bounds on the parameters* ω*,* λ *and* τ*, and lower bounds on* k*, such that, for all sufficiently large times* t*, the team-players' joint behavior at* t *constitutes an* ϵ*-Nash equilibrium of the equivalent game* $\hat{\mathbf{G}}$ *with probability at least* $1 - \epsilon$*.*

Note that an ϵ-Nash equilibrium are not necessarily the optimal equilibrium of the game $\hat{\mathbf{G}}$.

4.7.3 Simulations

This section presents simulation results of the micro-player matching with a de Bruijn ID sequence. Consider a team-based zero-sum game $\mathbf{G}$. In the simulation, the action sets were $X^1 = X^2 = Y = \{1, 2, 3\}$. Each of the team players $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ had $m = 3$ micro-players, and the opponent $\mathcal{P}_Y$ had a static bounded recall strategy with an order $k = 5$. The micro-player ID sequence s was selected from the set of de Bruijn sequences $\mathcal{B}(m, k+1)$. Therefore a period of action sequence was $L = m^{k+1} = 3^6 = 729$. Similar to the illustrative example in Section 4.3, the team players' payoff function $\phi : X^1 \times X^2 \times Y \rightarrow$

$\mathcal{R}$ is as follows. As shown in Table 3, if $x^1 = x^2 = y$ then $\phi(x^1, x^2, y)$ is 1, and otherwise $\phi(x^1, x^2, y)$ is 0.

Table 3: Payoff function $\phi(x^1, x^2, y)$ for simulations

y =**1**

x^2 \ x^1	**1**	**2**	**3**
1	1	0	0
2	0	0	0
3	0	0	0

y =**2**

x^2 \ x^1	**1**	**2**	**3**
1	0	0	0
2	0	1	0
3	0	0	0

y =**3**

x^2 \ x^1	**1**	**2**	**3**
1	0	0	0
2	0	0	0
3	0	0	1

The opponent $\mathcal{P}_Y$'s k-recall strategy $\sigma \in \Sigma_k$ was as follows. Every time step, the opponent $\mathcal{P}^Y$ selected an action $y \in Y$ that has the least empirical frequency in the last k-length history of team players' actions. If there exist more than one action with the least empirical frequency, one action among them was selected uniformly random. For examples, if $k = 5$ and the last 5-length action histories of x^1 and x^2 are '12213' and '23211' respectively, then $\mathcal{P}_Y$ selects 3 as his action.

One can easily prove that, for the aforementioned opponent's strategy σ, the optimal one-shot correlated joint strategy $\mu^*_{\text{cor}} \in \Delta(X^1 \times X^2)$ for the team is a uniform distribution over the three joint actions $\{(1,1),(2,2),(3,3)\}$. Then the expected reward for μ^*_{cor} is

$$\max_{\mu \in \Delta(X_1 \times X_2)} \phi(\mu, \sigma) = \phi(\mu^*_{\text{cor}}, \sigma) = \frac{1}{3}.$$

Therefore the average rewards in the simulations will be compared with $\frac{1}{3}$.

Note that, in Theorem 4.7.1, the team-player's joint behavior converges to a neighborhood of Nash equilibrium of the equivalent normal form game $\hat{\mathbf{G}} = (\{\mathcal{P}_X^{i,j}\}, \{X_i\}, \{U_{i,j}\})$. The following claim proves that an optimal joint action profile for micro-players is a Nash equilibrium.

Claim 4.7.1. *In the setup for simulations, every optimal joint action profile of micro-players is a Nash equilibrium of the game* $\hat{\mathbf{G}}$.

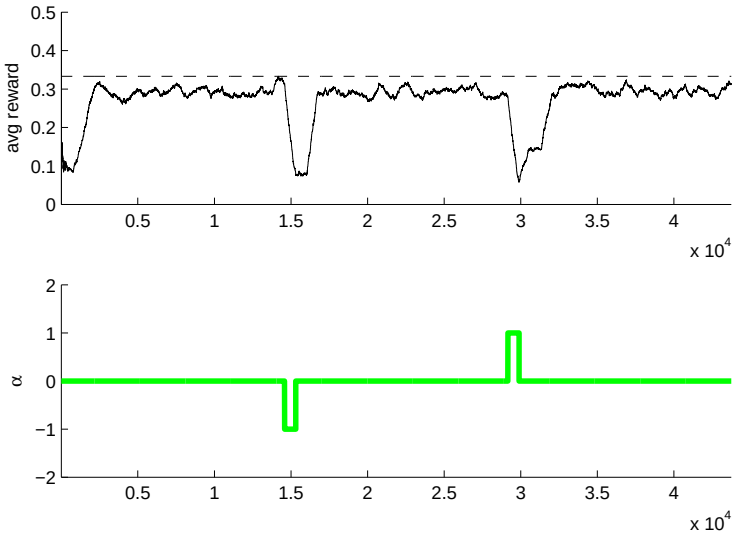


Figure 15: Simulation result of micro-player matching with a de Bruijn sequence. The upper graph shows the average reward and the lower one shows the phase difference α. The positive value of α means that $\mathcal{P}_X^1$ is lagging, and a negative value means that $\mathcal{P}_X^2$ is lagging. And α is zero when there is no phase difference between $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$.

Proof. Every optimal joint action sequence of the team players with $m = 3$ micro-players is one of de Bruijn sequences $\mathcal{B}(m, k+1)$ over the joint action set $\{(1,1),(2,2),(3,3)\}$. Consequently, each pair of micro-players $(\mathcal{P}_X^{1,j}, \mathcal{P}_X^{2,j})$ should coordinate their joint action to one of the joint action set $\{(1,1),(2,2),(3,3)\}$, and it should not be overlapped with other pair of micro-players. It is obvious that any unilateral deviation from such joint action profiles will damage the deviating micro-player's utility $U_{i,j}$. □

Figure 15 shows the simulation result. The upper graph in Figure 15 shows the average reward during the last $L = 729$ steps and the dotted line at $\frac{1}{3}$ is drawn for comparison purpose. The lower graph indicates a value of phase difference α between the team players ID sequences. The positive value of α means that $\mathcal{P}_X^1$ is lagging, and a negative value means that $\mathcal{P}_X^2$ is lagging. And α is zero when there is no phase difference between $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$. In the simulation, the parameters are set as $\omega = 0.1$, $\lambda = 0.01$, $\tau = 0.15$, and $\beta = 25$.

To test the phase lock process, the index of ID sequence s in a team player was artificially lagged by one at time $t = 14580$ and $t = 29169$ during the simulation. As shown in the second graph in Figure 15, artificially inserted phase differences were properly corrected in the simulation.

Furthermore, the average reward reached almost $\frac{1}{3}$ which is the expected reward for μ^*_{cor}. When there existed phase difference, the average reward went down but it was recovered soon after the team players' phases are locked again. Note that the average reward was little bit less than $\frac{1}{3}$ even though team players coordinated their action sequences. The reason is that micro-players experimented trial actions with probability w even after they optimized their actions. Therefore one can conclude that micro-player matching with a de Bruijn sequence could performed well as much as correlated one-shot strategies.

4.8 Summary and discussion

4.8.1 Summary

The goal of this chapter was to coordinate team players' actions without any communications in team-based zero-sum games. Generally, some global signalling devices are required for common randomness between players, but communications are very limited or impossible in many practical applications. Instead of learning a one-shot strategy, we let players coordinate a periodic sequence of deterministic actions and put an assumption on opponent's rationality. Since team players' action sequences are periodic and deterministic, common randomness is no more required to coordinate players. And it is proved that if a length of a periodic action sequence is long enough, then opponents with limited rationality cannot recognize its pattern. Because the opponents cannot recognize that the players are playing deterministic actions, the players' behavior looks like a correlated and randomized joint strategy with empirical distribution of their action sequences. Consequently players can coordinate their action sequences without any communication or global signals, and the average rewards of the resulted action sequences are as good as correlated joint strategies.

The notion of micro-players are introduced for efficient learning of long action sequences. By introducing a concept of micro-players, a problem of optimizing a parameter in a higher dimension is converted into an optimization of multiple parameters in a smaller dimension. In the first type of micro-player matching, a group of m micro-players within a player plays actions with a serial order ID sequence $s = (1, 2, ..., m)$.

In another micro-player matching method, in which a de Bruijn sequence is adopted as a ID sequence, players can generate very long action sequences without increasing the number of micro-players. The number of micro-players in this method can be independent from rationality level of the opponents. To correct phase difference between team players' action sequences, the uncertainty coefficients of realized rewards and shifted ID sequences are measured. The simulation results are shown to demonstrate the performance of micro-player matching methods.

4.8.2 Discussion

The proposed methods are constructed upon an important assumption that the team-players time steps are synchronized. We stated that micro-player matching does not require explicit global signals and communications between team players, but one might view '*time steps*' in our methods as the common signal to which the players could correlate their actions. Even though the assumption on synchronized time steps is a little bit relaxed by considering small phase differences between the team-players, if time steps for each team player in the system is not synchronized well then our methods cannot be applied.

In the de Bruijn ID sequence approach, the uncertainty coefficients were tested as a phase difference detection. The uncertainty coefficients could measure a level of association between realized rewards and shifted ID sequences $s^{\mathrm{sh}(-j)}$ with some small value j, because we assumed that phase difference rarely happened in the system. However exact limitation of the uncertainty coefficient as an association measurement is not completely analyzed yet.

Also the de Bruijn ID sequence approach has an optimality issue. In the first micro-player matching method with a serial order ID sequence, the original team-based zero-sum game can be converted to a potential game between the team players for a static opponent's strategy. Potential games are the special type of games in which the potential function maximizer is a Nash equilibrium and better-reply dynamics can reach the potential function maximizer. Consequently, existing learning algorithms for potential games, which are based on better-reply dynamics, can guarantee the convergence to the neighborhood of the optimal point. On the other hand, in the de Bruijn ID sequence approach, the team-based zero-sum game is converted to a normal form game between the team players. Since Theorem 2.3.3 showed that a regret testing rule converges to neighborhood of a Nash equilibrium of general finite normal form games, it can be applied as a learning algorithm for micro-players. However a Nash equilibrium of the game is not necessarily an optimal point, and thus the resulted behavior of micro-player matching with de Bruijn ID sequence may not be optimal. For better results, a proper equilibrium selection or micro-players' utility functions that induce a game with better properties have to be researched.

Lastly, we have assumed that the opponent player's strategy is static up to this point. In practical examples, it is more natural to think that opponent players also try to optimize their strategies even though their rationality is limited. If opponent players start to adapt their strategies, their strategies are time-varying and consequently, utility functions of micro-players like (8) and (14) are also time-varying functions. These time-varying utility functions are related to the results regarding stochastic stability in drifting environments from Chapter 3. More details about team-based zero-sum games with dynamically adapting opponents will be discussed in the next chapter.

CHAPTER 5

PAYOFF-BASED COORDINATION IN DRIFTING ENVIRONMENTS

This chapter combines the results of the previous two chapters by considering a problem of coordinating actions without communications in drifting environments. More specifically, it is assumed that the opponent player in the team-based zero-sum games from the previous chapter tries to adjust its strategy in the set of bounded recall strategies.

The first section analyses the team-based zero-sum games with adaptive opponents and some simulation results are shown. In the second section, we developed the human testbed program for further study regarding a human as an adaptive opponent in the team-based zero-sum games. A user of the human testbed program is supposed to play series of stages of the zero-sum game against a team of virtual players.

5.1 Team-based zero-sum games with adaptive opponents

Consider a team-based zero-sum game $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ described in Chapter 4 again. In the previous chapter, it was assumed that the opponent player $\mathcal{P}_Y$'s bounded recall strategy $\sigma \in \Sigma_k$ was static, and consequently the team players' average payoff function ϕ_σ^L in (7) was also static. A static strategy σ means that the opponent player never changes his strategy during run time. However in many practical applications, it is more natural to think that the opponent player also tries to learn and adapt his strategy.

Assume that the opponent adapts his strategy σ within a bounded recall strategy set Σ_k. Let $\sigma_t \in \Sigma_k$ be the opponent's strategy at time t. Suppose that the opponent's adapting speed is slow, i.e., $|\sigma_t - \sigma_{t-1}|$ is bounded with some small value regardless a method used to adapt the opponent's strategy. Then one can easily recognize the connection between this example and Theorem 3.2.2 in Chapter 3. Note that in Lemma 4.6.1 and Lemma 4.7.1,

the game $\mathbf{G}$ has a corresponding normal form game $\hat{\mathbf{G}} = (\{\mathcal{P}_X^{i,j}\}, \{X_i\}, \{U_{i,j}^{\sigma}\})$ of micro-players. Similarly, for every opponent's strategy $\sigma_t \in \Sigma_k$, we will denote the equivalent normal form game of micro-players as $\hat{\mathbf{G}}_t$. Then the slowly varying opponent's strategy σ in $\hat{\mathbf{G}}_t$ plays the same role as the environment parameter θ in Theorem 3.2.2. The following corollaries are direct consequences of Chapter 3.

Corollary 5.1.1. *Consider a team-based zero-sum game* $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$. *Suppose that the team players* $\{\mathcal{P}_X^1, \mathcal{P}_X^2\}$ *play the micro-player matching with either a serial ID sequence or a de Bruijn ID sequence. Assume that* $\mathcal{P}_Y$ *plays a bounded recall strategy* $\sigma_t \in \Sigma_k$ *at time* t. *Then for every* ϵ, *there exist* $\delta > 0$ *and learning parameters such that*

$$|\sigma_t - \sigma_{t-1}| < \delta$$

for all $t = 0, 1, 2...$ *implies that the team-players joint behavior at* t *constitutes an* ϵ*-Nash equilibrium of the the equivalent game* $\hat{\mathbf{G}}_t$ *of micro-players with probability at least* $1 - \epsilon$.

Proof. Team players with micro-player matching update their action sequences every end of action sequence period, hence one can consider the system as a perturbed Markov process evolving period-by-period over the set of joint action sequences, $\prod_{l=1}^{L} X$. For each opponent's static strategy $\sigma \in \Sigma_k$, let P_σ^ϵ be a Markov process induced by the micro-player matching. Define $\mu_\sigma^* \in \Delta(\prod_{l=1}^{L} X)$ as the associated stationary distribution over the finite set of L-length joint action sequence. Then the rest of proof is easily induced from the results of the previous chapters. □

5.1.1 Simulations

This section presents simulations of the micro-player matching against an adapting opponent. Most of simulation setting is similar to Section 4.7.3 except the opponent player. Consider a team-based zero-sum game $\mathbf{G} = (\{\mathcal{P}_X^1, \mathcal{P}_X^2, \mathcal{P}_Y\}, \{X_1, X_2, Y\}, \phi)$ where the action sets for players were $X^1 = X^2 = Y = \{1, 2, 3\}$. The team players' payoff function $\phi : X^1 \times X^2 \times Y \to \mathcal{R}$ is 1 if $x^1 = x^2 = y$, and 0 otherwise. The opponent $\mathcal{P}_Y$ has

a time-varying bounded recall strategy with order $k = 4$, i.e., $\sigma \in \Sigma_4$. Each of the team players $\mathcal{P}_X^1$ and $\mathcal{P}_X^2$ had $m = 3$ micro-players, and the micro-player ID sequence s was selected from the set of de Bruijn sequences $\mathcal{B}(m, k+1) = \mathcal{B}(3, 5)$.

Define $h_k(t) \in \prod_{l=1}^{k} X$ as a k-length joint history of the team players at time t, i.e.,

$$h_k(t) = (x(t-k), x(t-k+1), ..., x(t-1)).$$

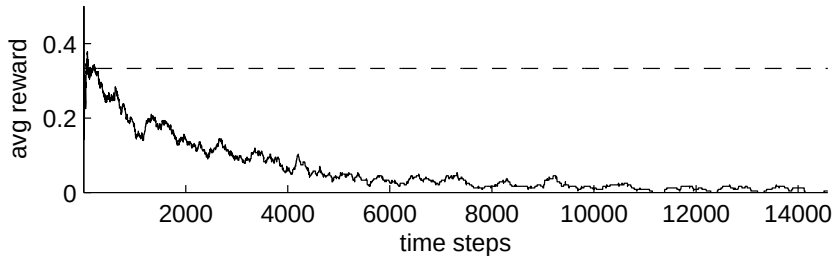
Every time step t, $\mathcal{P}_Y$ selects its action $y(t) \in Y$ according to its bounded recall strategy $\sigma_t : \prod_{l=1}^{k} X \rightarrow \Delta(Y)$. In other words, $\mathcal{P}_Y$ assigns a probability distribution $\sigma_t(h_k) \in \Delta(Y)$ to every available k-length joint action history $h_k \in \prod_{l=1}^{k} X$ of the team players. Based on the observed joint action profile $x(t)$, the opponent's strategy σ_t is updated with the following rule.

$$\sigma_{t+1}(h_k(t)) \leftarrow \alpha \cdot v_{x(t)} + (1-\alpha) \cdot \sigma_t(h_k(t))$$
$$\sigma_{t+1}(h'_k) \leftarrow \sigma_t(h'_k), \qquad \forall h'_k \neq h_k(t)$$

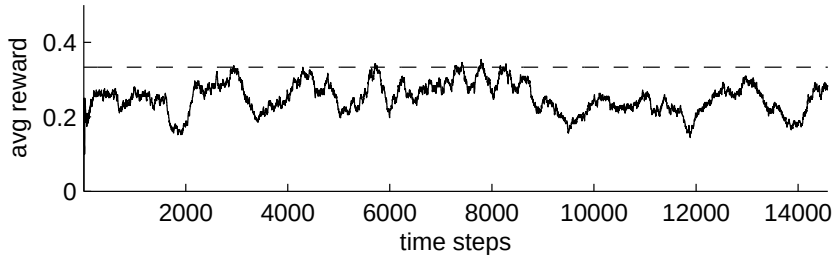
where $\alpha \in [0, 1]$ is a learning rate and $v_{x(t)} \in \Delta(Y)$ is a uniform distribution over the set $Y \setminus \{x^1(t), x^2(t)\}$. In other words, the opponent adapts his strategy by decreasing the probability of the actions played by the team players. In simulations, the opponent's adapting speed $|\sigma_{t+1} - \sigma_t|$ is bounded by limiting the parameter α to be small.

Figure 16 shows the simulation results of the team-based zero-sum game against the adaptive opponent illustrated above. First, Figure 16 (a) shows the opponent's ability to adapt his strategy against static team players. In this simulation, the team players played the same static and periodic action sequence and did not update it during the simulation. The result shows that the opponent could successfully optimize his strategy so that the average reward of the team players converges to zero after sufficiently large time.

Next, the simulation result of the micro-player matching method against the adaptive opponent is shown in Figure 16 (b). Contrary to the simulation result in Figure 16 (a), the team players could adapt their action sequences against the opponent who adapted its strategy dynamically. The average rewards of the team were maintained in some neighborhood



(a) team players with a static and periodic joint action sequence against the adapting opponent



(b) team players with the micro-player matching algorithm against the adapting opponent

Figure 16: Team players' average rewards against the adapting opponent. The dotted lines are drawn at $\frac{1}{3}$ which is a maxmin reward of the correlated equilibrium of the game. With the micro-player matching method, the team players could adapt their action sequences against the opponent who adapted its strategy dynamically.

of $\frac{1}{3}$ which was a maxmin reward of the correlated equilibrium of the game. Hence the simulation results indicates that the micro-player matching works against a opponent who dynamically adapts its strategy with sufficiently slow speed.

5.2 Micro-player matching against humans

In this section, a graphical user interface (GUI) human testbed program is developed to experiment the micro-player matching against humans. A real person will act and make choice as an opponent with limited rationality in team-based zero-sum games.

Rationality of humans has been a popular research topic in the behavioural and social sciences, and 'rationality' can be viewed as an aspect of reasoning in decision making [73]. Even though the decision making process of human is not completely understood yet, large empirical evidence of human behaviour showed that human rationality is not perfect. In economics which is a largely model-based science, lots of models of rational choice take

into account limitations of human capacities as bounded rationality [74]. Referring to the citation [73], some typical assumptions on the limitation of human abilities in economy theory are as follows; i) limited knowledge of the world, ii) limited ability to evoke this knowledge, iii) limited ability to cope with uncertainty, and iv) limited ability to evoke possible courses of action.

Even though we cannot argue that bounded recall strategy is the best model for rationality of humans, we believe that testing micro-player matching algorithm against a human is still a meaningful test. One can find the similarity between the bounded rationality model and limited rationality of human in the sense that people apparently have limited memory when trying to memorize a long sequence of symbols.

5.2.1 Human testbed program

A simple human testbed program (see Figure 17) is developed using MATLAB for the experiments. The team-based zero-sum game $\mathbf{G}$ applied for the testbed is similar to the illustrative example in Chapter 4.3. All players, including a human participant and two virtual team players, have the same action set $\{left, right\}$. A human participant loses the stage if all three players choose the same action, or wins the stage otherwise. The game is repeated until a user stops the program.

The testbed program consists of the following functions.

- Play a stage by selecting an action among $\{left, right\}$
- Reset all history and parameters
- Save the current setting and play history as a separate file
- Quit the program

The program displays the current time steps (i.e., the stage number) and the participant's average winning rate during the test. A user can select an action by pressing a left or right arrow key on a keyboard, or clicking a button on the screen. Whenever a user selects an

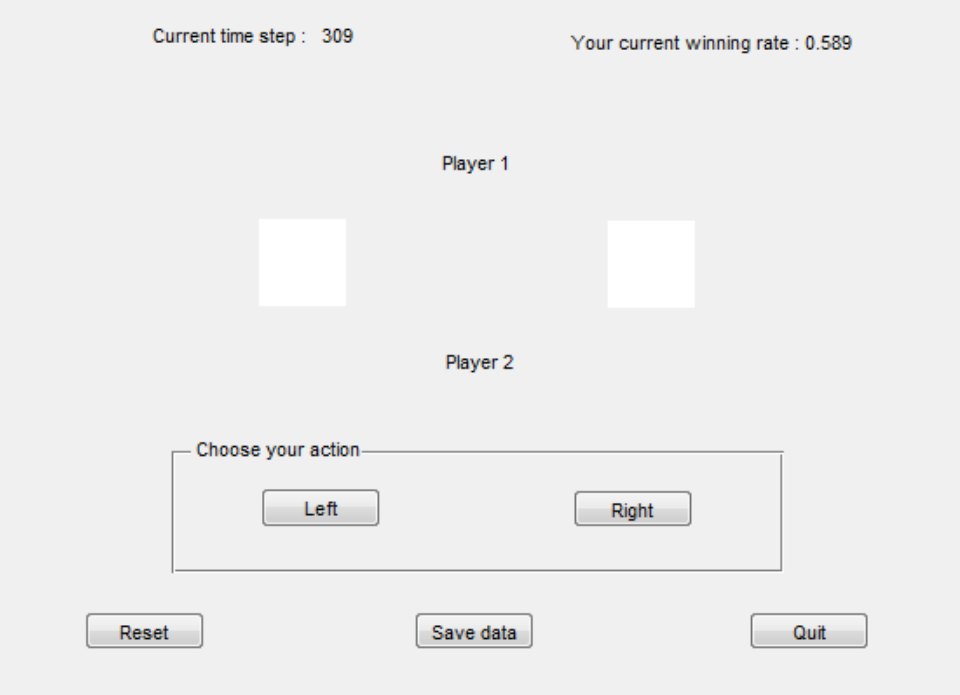
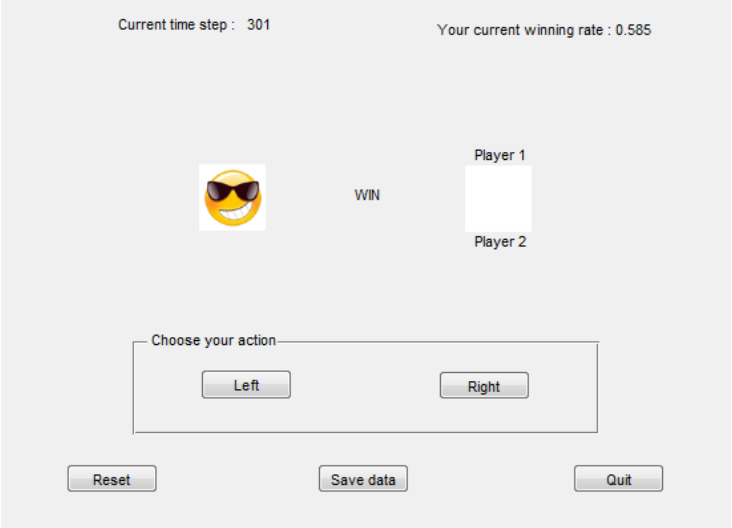


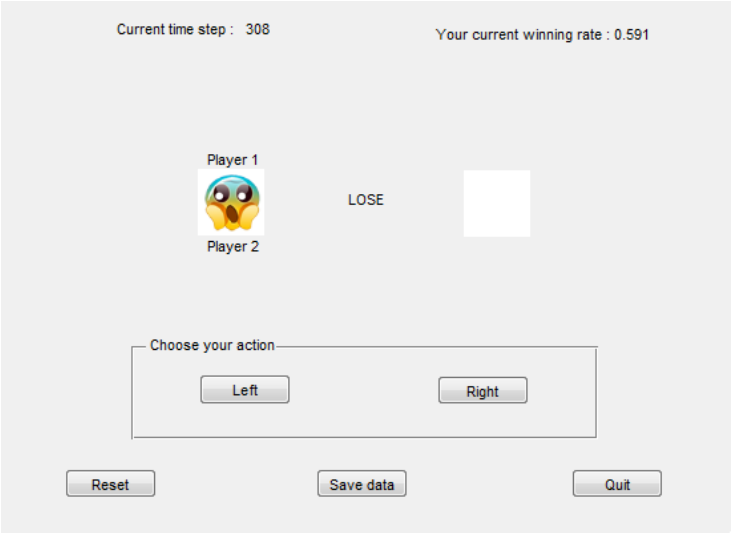
Figure 17: Human testbed program. Both *Player 1* and *Player 2* are at a neutral position which is the middle of the *left* and *right* boxes while waiting for the user's input. Once the user selects his action, then each *Player 1* and *Player 2* moves to one of the *left* and *right* boxes according to their actions.

action, the virtual team players, *Player1* and *Player2*, immediately select their own actions using the micro-player matching, then the result is visually displayed on the program. Then for a user's intuitive recognition, the result whether the user won or lost the stage is shown in the box selected by the user. Figure 18 shows the sample screen shots when the user wins or loses a stage against the team players.

In the human testbed program, it is important to make sure that the team players' action sequences have an appropriate length. Testing with impractically long action sequences (e.g. sequence with more than 30-length period) against a human participant will take too much time. Moreover there is a risk that experiment participants might be overwhelmed by the length of the sequence and give up to adapt his actions. On the other hand, if the sequence length is too short then the game is too trivial for participants. Hence the sequence length L was selected as a random number between 9 to 13, and the selected



(a) Example when a user won the stage. The user chose *'left'* while the virtual team-players chose *'right'*



(b) Example when a user lost the stage. All players including the user and virtual team-players chose *'left'*

Figure 18: Screenshots of the program when a user won and lost the stage. A picture with facial expression helps the user to intuitively recognize a result of each stage.

length is unknown to the user.

The following information about the system was informed to participants before experiments.

- Game rules such as the available action set and condition for winning the stage
- Each team player repeats its own action sequence while gradually adapts it with some

algorithm.

- The period of the team players' action sequences is randomly selected between 9 to 13 at the start of the test.
- The team players cannot observe the participant's action or its team-mate's action.
- The team players can only observe the result whether they won or lost the stage.

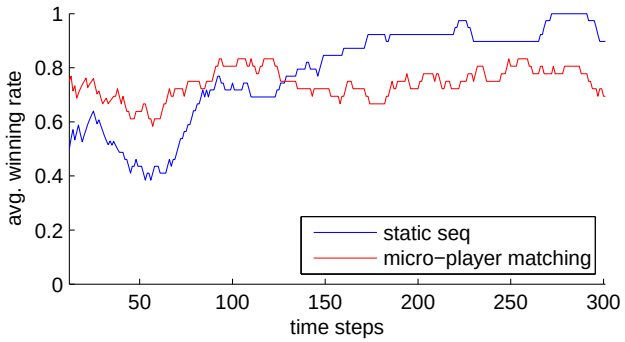
5.2.2 Test results

To demonstrate the developed human testbed program, this section presents brief analysis of the test results with experiment participants. Note that the purpose of this section is not exhaustive study about correlated behavior learning against humans. However the developed testbed program can be utilized for the further in-depth research about the topic.

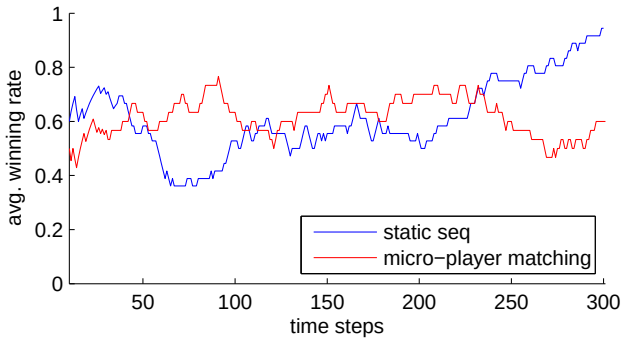
Experiments were performed with three participants and each participant played two sessions of the game. Each session was conducted for 300 stages. In the first session, the team players repeated static action sequences without any learning algorithm. Both players had the same static action sequence so that the team players always selected the same action with a prescribed order. The static action sequences were randomly selected during the initialization before starting the experiments. In this case, if the experiment participant could memorize the sequence perfectly then they could always win the game against the virtual team players.

In the second session, the team players adapted their action sequences using the micro-player matching with a serial ID sequence illustrated in Section 4.6.1. The initial action sequence was independently selected for each player so that the team players' actions were not coordinated at the start of the experiments. As explained in the previous section, the sequence length L was a random number between 9 to 13 and unknown to the participants.

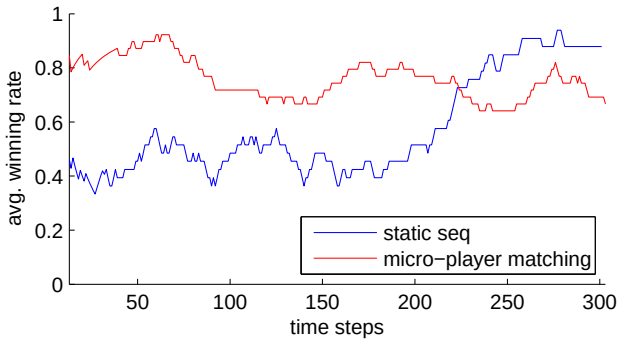
The experiment results are shown in Figure 19. Because the static action sequence given to each participant was randomly selected, the pattern of the static action sequence for a specific participant could be more complex than others. Hence a participant might feel



(a) Participant 1



(b) Participant 2



(c) Participant 3

Figure 19: The human testbed program results with three experiment participants. In each graph, the blue line and red line are the average winning rates of the participant against a static action sequence and a team of players with micro-player matching method respectively.

it was trickier to remember the sequence than other participants during the early stages. Nonetheless, it was commonly observed in all participant's test results that the winning rate against a static action sequence converged to almost one at the end of the session.

On the other hand, the winning rates against the micro-player matching method were not improved during the session for all participants. One can interpret the results as that the micro-player matching method could successfully hamper the human participant's learning.

5.3 Summary

This chapter presented coordination with no communications in drifting environments by considering team-based zero-sum games in which an opponent dynamically adapts its strategy. It is assumed that the opponent player's learning speed is slow enough so that its strategy is gradually changed within a bounded recall strategy set. Then the team of players can learn correlated behavior with the micro-player matching method.

Moreover, a GUI human testbed program is developed to experiment the micro-player matching against humans. Even though the exact model for the rationality of humans is not understood yet, there is some similarity between the bounded recall strategy and limited rationality of human. A thorough research regarding game theoretic learnings against human is beyond the scope of this research. Nonetheless, we believe that testing micro-player matching algorithm against a human is an interesting topic and the test results presented in this chapter suggest the future research direction.

CHAPTER 6

CONCLUSION AND FUTURE DIRECTIONS

The goal of this dissertation is to analyze and develop game theoretic distributed control under dynamic and challenging conditions: drifting environments and constrained communications. In this chapter, we will conclude our research regarding the two challenging settings then discuss future research directions.

6.1 Robustness of stochastic stability

The effects of two types of environmental disturbances have been analyzed in terms of stochastic stability which is a notion of convergence in game theoretic learning. The first setup involves perturbed dynamics. One motivation in the game theoretic learning framework stems from a presumed ability to measure utility functions. These computations may require estimates from persistently noisy measurements or may neglect latent state variables. We establish a continuity result that shows how small perturbations to the learning rule induce small effects on the limiting stochastic behaviors. The second setup concerns slowly varying dynamics, i.e., drifting environments. A standard assumption in game theoretic learning is a stationary environment, e.g., the game is fixed. We investigate the case of slow variations and show that for sufficiently slow time variations, the limiting behavior "tracks" the stochastically stable states. Since the analysis s regarding Markov processes, the results could be applied to various game theoretic learning rules. In this thesis, the results were applied to log-linear learning and the mobile sensor coverage example was tested in both simulation and laboratory experiments.

6.2 Correlated behavior in payoff-based learning

In the second part of this dissertation, we researched a payoff-based correlated behavior learning under the presence of hostile opponents. It was assumed that team players have

limited observability, and an opponent has limited rationality. The task for the team players is to learn correlated behavior without communications while not disclosing their behavior to the opponent.

Achieving correlated behavior is one of the major concerns in distributed learnings, and it generally requires some common signals that can be observed by all players. However, in completely uncoupled settings, common signals or communications are impossible for a group of players so that much research has only concerned about independently mixed strategies. On the other hand, we considered that 'time' is a common signal that implicitly always shared by all players in repeated game settings. Players can correlate their actions to time steps by playing deterministic and periodic action sequences instead of randomized one-shot strategies. It is shown that the length of sequence is related to the level of opponent's rationality.

We have introduced a notion of micro-players to improve periodic action sequence learning. Then adapting a joint action sequence is interpreted as matching micro-players in each team player. Moreover by introducing a de Bruijn sequence, we could successfully separate the level of opponent's rationality and the size of the game of micro-players.

The micro-player matching method provides a new frame that converts the original team-based zero-sum game to a game between micro-players. In our research, we considered specific existing algorithms such that log-linear learning and regret testing as learning algorithms for each micro-player. But one may apply other existing payoff-based learnings and proper utility designs to achieve better equilibrium selection and/or learning rate.

6.3 Future directions

Future research for game theoretic learning in drifting environments includes application to distributed robotics systems. Also one can develop a model of the environmental variations to help reduce the time scale separation required in this approach.

For the micro-player matching method, future work should focus on in-depth research

about the learning rate of the proposed methods and phase-locking process that is robust to larger phase differences In the de Bruijn ID sequence approach, the uncertainty coefficient was utilized for phase-locking process between team players. However, the exact limitations and characteristics of the uncertainty coefficient as an association measurement is not completely analyzed yet. Payoff-based phase-locking mechanism with the uncertainty coefficient should be explored in future.

Also the de Bruijn ID sequence approach has an optimality issue. A regret testing rule applied as a micro-player's learning algorithm converges to neighborhood of a Nash equilibrium of a normal form game between micro-players. However a Nash equilibrium is not necessarily an optimal point of the game, and thus the resulted behavior of micro-player matching with de Bruijn ID sequence may not be optimal. For better results, a proper equilibrium selection or micro-players' utility functions that induce a game with better properties have to be researched.

Lastly, the developed GUI human testbed program can be a starting point for studying game theoretic correlated behavior learning against a human.

REFERENCES

[1] R. Gopalakrishnan, J. R. Marden, and A. Wierman, "An architectural view of game theoretic control," *SIGMETRICS Perform. Eval. Rev.*, vol. 38, pp. 31–36, Jan. 2011.

[2] D. Fudenberg and D. Levine, *The Theory of Learning in Games*. Cambridge, MA: MIT Press, 1998.

[3] S. Hart, "Adaptive heuristics," *Econometrica*, vol. **73**, no. 5, pp. 1401–1430, 2005.

[4] H. P. Young, *Strategic Learning and its Limits*. Oxford University Press, 2005.

[5] J. R. Marden, G. Arslan, and J. S. Shamma, "Connections between cooperative control and potential games," *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, vol. 39, pp. 1393–1407, December 2009.

[6] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 45, pp. 436 – 463, 2005.

[7] J. R. Marden, S. Ruben, and L. Pao, "Surveying game theoretic approaches for wind farm optimization," in *Proceedings of the AIAA Aerospace Sciences Meeting*, January 2012.

[8] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: a game theoretical formulation," *ASME Journal of Dynamic Systems, Measurement and Control*, pp. 584–596, September 2007.

[9] E. Campos-Naòez, A. Garcia, and C. Li, "A game-theoretic approach to efficient power management in sensor networks," *Oper. Res.*, vol. 56, pp. 552–561, May 2008.

[10] J. Neel, A. Mackenzie, R. Menon, L. Dasilva, J. Hicks, J. Reed, and R. Gilles, "Using game theory to analyze wireless ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. **7**, pp. 46–56, 2005.

[11] L. Blume, "The statistical mechanics of strategic interaction," *Games and Economic Behavior*, vol. 5, pp. 387–424, July 1993.

[12] J. R. Marden and J. S. Shamma, "Revisiting log-linear learning: Asynchrony, completeness, and payoff-based implementation," *Games and Economic Behavior*, vol. 75, pp. 788–808, 2012.

[13] J. R. Marden, "State based potential games," *Automatica*, vol. 48, no. 12, pp. 3075–3088, 2012.

[14] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[15] D. Monderer and L. S. Shapley, "Potential games," *Games and Economic Behavior*, vol. 14, pp. 124–143, May 1996.

[16] J. R. M. R. Gopalakrishnan and A. Wierman, "Potential games are necessary to ensure pure nash equilibria in cost sharing games." under submission, 2013.

[17] J. R. Marden, G. Arslan, and J. S. Shamma, "Joint strategy fictitious play with inertia for potential games," *IEEE Transactions on Automatic Control*, vol. 54, pp. 208–220, February 2009.

[18] J. S. Shamma and G. Arslan, "Dynamic fictitious play, dynamic gradient play, and distributed convergence to nash equilibria," *Automatic Control, IEEE Transactions on*, vol. 50, no. 3, pp. 312–327, 2005.

[19] J. R. Marden, H. P. Young, G. Arslan, and J. S. Shamma, "Payoff based dynamics for multi-player weakly acyclic games," *SIAM Journal on Control and Optimization*, vol. 48, pp. 373–396, February 2009.

[20] H. P. Young, "The evolution of conventions," *Econometrica*, vol. 61, pp. 57–84, January 1993.

[21] J. R. Marden, G. Arslan, and J. S. Shamma, "Regret based dynamics: convergence in weakly acyclic games," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, AAMAS '07, (New York, NY, USA), pp. 42:1–42:8, ACM, 2007.

[22] D. H. Wolpert, K. Wheeler, and K. Tumer, "Collective intelligence for control of distributed dynamical systems," *Europhysics Letters*, vol. 49, March 2000.

[23] H. P. Young, *Individual Strategy and Social Structure*. Princeton, NJ: Princeton University Press, 1998.

[24] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*. Athena Scientific, 2002.

[25] J. W. Weibull, "Evolutionary game theory," *MIT Press Books*, vol. 1, 1997.

[26] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[27] L. Samuelson, *Evolutionary games and equilibrium selection*, vol. 1. Mit Press, 1998.

[28] D. Shah and J. Shin, "Dynamics in congestion games," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 38, pp. 107–118, ACM, 2010.

[29] D. P. Foster and H. P. Young, "Regret testing: learning to play nash equilibrium without knowing you have an opponent," *Theoretical Economics*, vol. 1, no. 3, pp. 341–367, 2006.

[30] H. P. Young, "Learning by trial and error," *Games and Economic Behavior*, vol. 65, no. 2, pp. 626–643, 2009.

[31] F. Germano and G. Lugosi, "Global nash convergence of foster and young's regret testing," *Games and Economic Behavior*, vol. 60, no. 1, pp. 135–154, 2007.

[32] M. Kandori, G. Mailath, and R. Rob, "Learning, mutation, and long-run equilibria in games," *Econometrica*, vol. 61, pp. 29–56, 1993.

[33] M. J. Fox and J. S. Shamma, "Communication, convergence, and stochastic stability in self-assembly," in *49th IEEE Conference on Decision and Control*, pp. 7245–7250, December 2010.

[34] M. J. Fox and J. S. Shamma, "Language evolution in finite populations," in *50th IEEE Conference on Decision and Control*, pp. 4473–4478, December 2011.

[35] Y. Lim and J. S. Shamma, "Robustness of stochastic stability in game theoretic learning," in *American Control Conference (ACC)*, 2013.

[36] O. Candogan, A. Ozdaglar, and P. Parrilo, "Dynamics in near-potential games." 2011.

[37] M. F. Balcan, F. Constantin, and S. Ehrlich, "The snowball effect of uncertainty in potential games," in *7th Workshop in Internet and Network Economics (WINE2011)*, December 2011.

[38] C. A.-F. andN. Netzer, "The logit-response dynamics," *Games and Economic Behavior*, vol. 68, no. 2, pp. 413–427, 2010.

[39] C. A.-F. andN. Netzer, "Robust stochastic stability." Working Paper No. 63, Department of Economics, University of Zurich, 2012.

[40] J. C. D. Meyer, "The condition of a finite markov chain and perturbation bounds for the limiting probabilities," *SIAM Journal on Algebraic and Discrete Methods*, vol. 1, no. 3, pp. 273–282, 1980.

[41] W. J. Rugh and J. S. Shamma, "A survey of research on gain scheduling," *Automatica*, pp. 1401–1425, 2000.

[42] J. S. Shamma and M. Athans, "Gain scheduling: Potential hazards and possible remedies," *IEEE Control Systems Magazine*, vol. **12**, no. 3, pp. 101–107, 1992.

[43] J. S. Shamma, "An overview of LPV systems," in *Control of Linear Parameter Varying Systems with Applications* (J. Mohammadpour and C. Scherer, eds.), pp. 3–26, Springer, 2012.

[44] url=http://www.k-team.com/mobile-robotics-products/.

[45] url=http://www.naturalpoint.com/optitrack/.

[46] M. G. Jones, "Design and implementation of a multi-agent systems laboratory," Master's thesis, Georgia Institute of Technology, 2009.

[47] Y. Lim, "Potential game based cooperative control in dynamic environments," Master's thesis, Georgia Institute of Technology, 2011.

[48] B. von Stengel and D. Koller, "Team-maxmin equilibria," *Games and Economic Behavior*, vol. 21, no. 12, pp. 309 – 321, 1997.

[49] R. J. Aumann, "Subjectivity and correlation in randomized strategies," *Journal of Mathematical Economics*, vol. 1, pp. 67–96, March 1974.

[50] F. Forges, "An approach to communication equilibria," *Econometrica: Journal of the Econometric Society*, pp. 1375–1385, 1986.

[51] O. Gossner and T. Tomala, "Secret correlation in repeated games with imperfect monitoring.," *Math. Oper. Res.*, vol. 32, no. 2, pp. 413–424, 2007.

[52] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, 2000.

[53] A. Cahn, "General procedures leading to correlated equilibria," *International Journal of Game Theory*, vol. 33, no. 1, pp. 21–40, 2004.

[54] D. P. Foster and R. V. Vohra, "Calibrated learning and correlated equilibrium," *Games and Economic Behavior*, vol. 21, no. 1, pp. 40–55, 1997.

[55] D. P. Foster and R. Vohra, "Regret in the on-line decision problem," *Games and Economic Behavior*, vol. 29, no. 1-2, pp. 7–35, 1999.

[56] J. R. Marden, "Selecting efficient correlated equilibria through distributed learning," 2013. submitted for publication.

[57] E. Ben-Porath, "Repeated games with finite automata," *Journal of Economic Theory*, vol. 59, pp. 17–32, February 1993.

[58] A. Neyman, "Cooperation, repetition, and automata," in *Cooperation: Game Theoretic Approaches, NATO ASI Series F*, pp. 233–255, Springer-Verlag, 1997.

[59] A. Neyman and D. Okada, "Two-person repeated games with finite automata," *International Journal of Game Theory*, vol. 29, no. 3, pp. 309–325, 2000.

[60] O. Gossner and P. Hernndez, "On the complexity of coordination," *Mathematics of Operations Research*, vol. 28, no. 1, pp. 127–140, 2003.

[61] R. Peretz, "Correlation through bounded recall strategies," *International Journal of Game Theory*, pp. 1–24, 2012.

[62] E. Lehrer, "Repeated games with stationary bounded recall strategies," *Journal of Economic Theory*, vol. 46, no. 1, pp. 130–144, 1988.

[63] C. H. Papadimitriou and T. Roughgarden, "Computing correlated equilibria in multi-player games," *J. ACM*, vol. 55, pp. 14:1–14:29, Aug. 2008.

[64] N. G. de Bruijn, "A Combinatorial Problem," *Koninklijke Nederlandsche Akademie Van Wetenschappen*, vol. 49, pp. 758–764, June 1946.

[65] A. Klein, *Stream Ciphers*. Springer, 2013.

[66] S. W. Golomb, *Shift Register Sequences*. San Francisco, CA: Holden-Day, 1967.

[67] C. Ronse, *Feedback Shift Registers*. Berlin: Springer-Verlag, 1984.

[68] A. Ralston, "De bruijn sequences-a model example of the interaction of discrete mathematics and computer science," *Mathematics Magazine*, pp. 131–143, 1982.

[69] K. Mandal and G. Gong, "Cryptographically strong de bruijn sequences with large periods," in *Selected Areas in Cryptography*, pp. 104–118, Springer, 2013.

[70] F. Annexstein, "Generating de bruijn sequences: An efficient implementation," *IEEE Transactions on Computers*, vol. 46, no. 2, pp. 198–200, 1997.

[71] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2006.

[72] H. Theil, "On the estimation of relationships involving qualitative variables," *American Journal of Sociology*, vol. 76, no. 1, pp. pp. 103–154, 1970.

[73] H. Simon, "Bounded rationality in social science: Today and tomorrow," *Mind & Society*, vol. 1, no. 1, pp. 25–39, 2000.

[74] T. Grüne-Yanoff, "Bounded rationality," *Philosophy Compass*, vol. 2, no. 3, pp. 534–563, 2007.